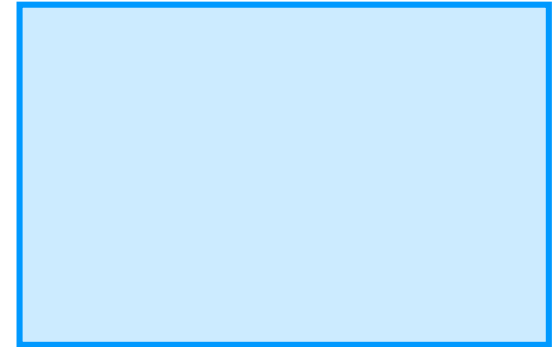


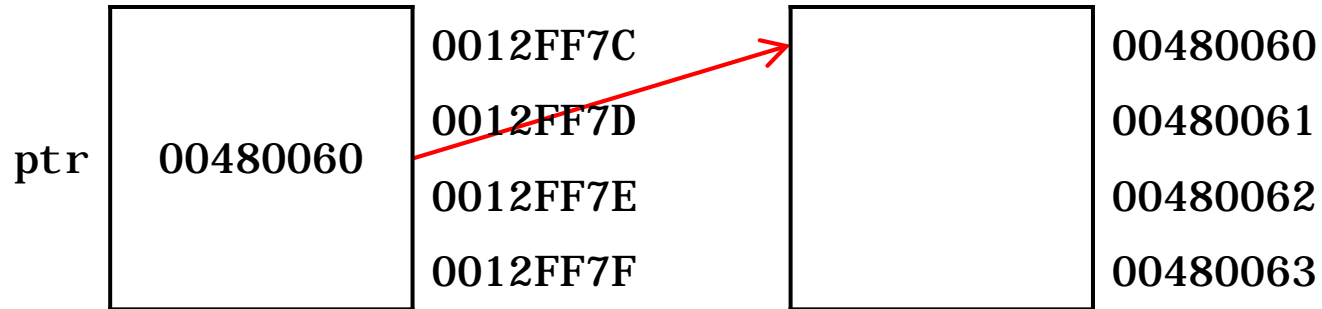
Dynamically allocate a variable



```
int main()
{
    // int *ptr = new int;
    int *ptr;
    ptr = new int;
    *ptr = 20;
    cout << *ptr << endl;
    cout << ptr << endl << endl;
    delete ptr;
}
```

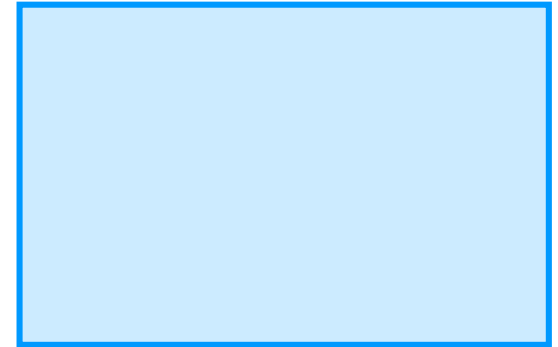
# Output

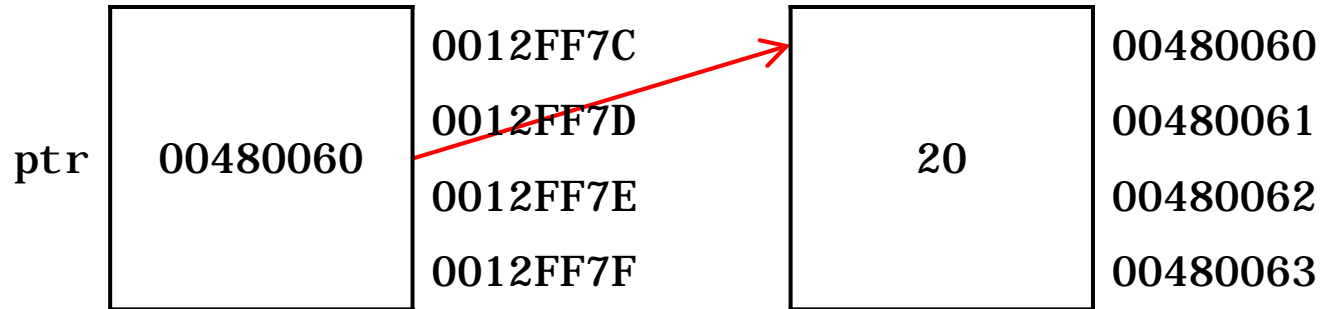




```
int main()
{
    // int *ptr = new int;
    int *ptr;
    ptr = new int;
    *ptr = 20;
    cout << *ptr << endl;
    cout << ptr << endl << endl;
    delete ptr;
}
```

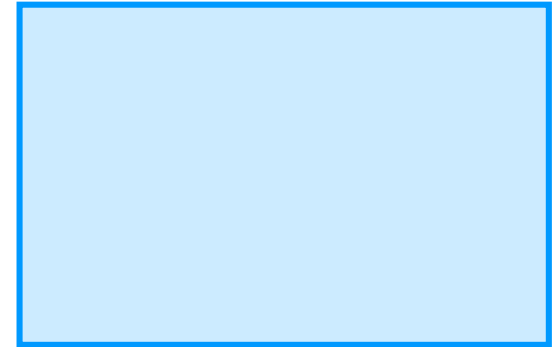
Output

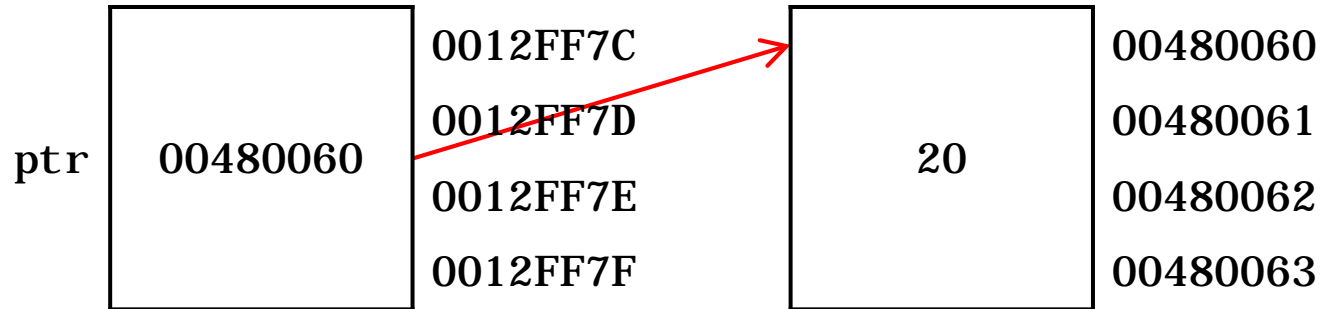




```
int main()
{
    // int *ptr = new int;
    int *ptr;
    ptr = new int;
    *ptr = 20;
    cout << *ptr << endl;
    cout << ptr << endl << endl;
    delete ptr;
}
```

Output

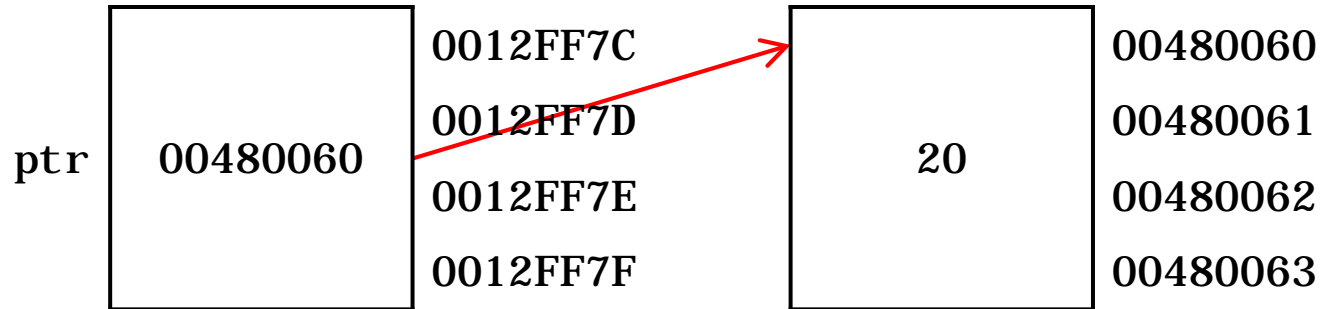




```
int main()
{
    // int *ptr = new int;
    int *ptr;
    ptr = new int;
    *ptr = 20;
    cout << *ptr << endl;
    cout << ptr << endl << endl;
    delete ptr;
}
```

# Output

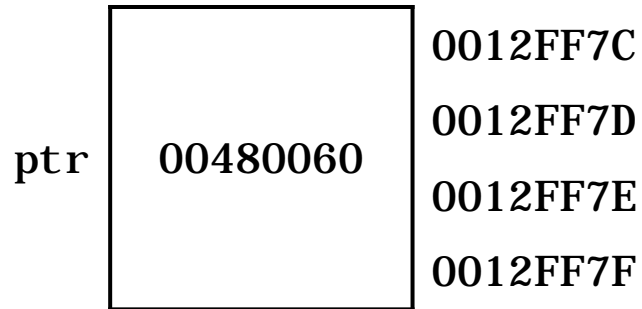
20



```
int main()
{
    // int *ptr = new int;
    int *ptr;
    ptr = new int;
    *ptr = 20;
    cout << *ptr << endl;
    cout << ptr << endl << endl;
    delete ptr;
}
```

# Output

```
20
00480060
```



```
int main()
{
    // int *ptr = new int;
    int *ptr;
    ptr = new int;
    *ptr = 20;
    cout << *ptr << endl;
    cout << ptr << endl << endl;
    delete ptr;
}
```

# Output

20
00480060

Dynamically allocate an array  
- Fibonacci numbers



fib		0012FF78
number		0012FF7C

```

int main()
{
    int number;
    cin >> number;
    int *fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```

fib		0012FF78
number		0012FF7C

```

int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```

fib		0012FF78
number	5	0012FF7C

```

int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```

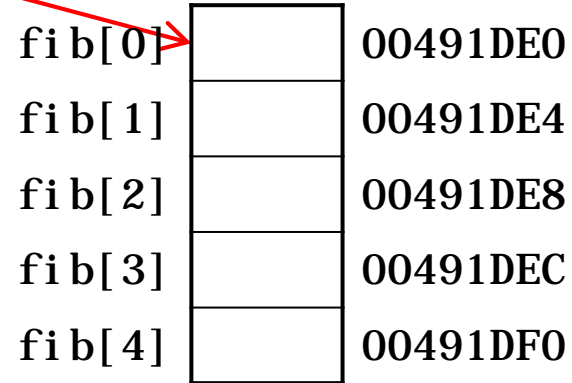
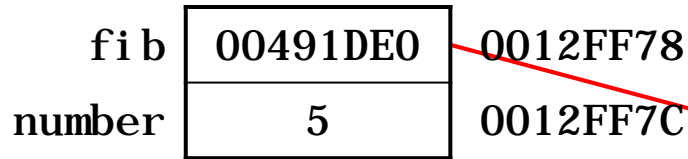
fib		0012FF78
number	5	0012FF7C

	00491DE0
	00491DE4
	00491DE8
	00491DEC
	00491DF0

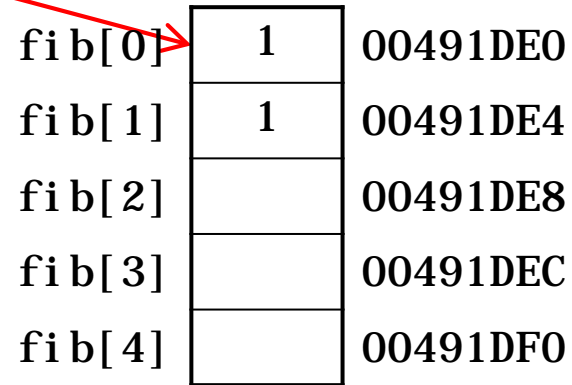
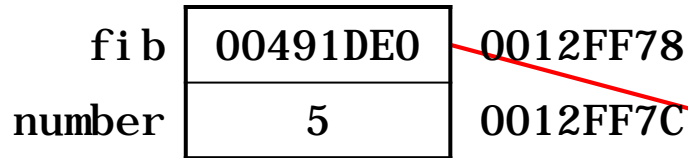
```

int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```



```
int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```

int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```

fib	00491DE0	0012FF78
number	5	0012FF7C

fib[0]	1	00491DE0
fib[1]	1	00491DE4
fib[2]	2	00491DE8
fib[3]		00491DEC
fib[4]		00491DF0

```

int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```

fib	00491DE0	0012FF78
number	5	0012FF7C

fib[0]	1	00491DE0
fib[1]	1	00491DE4
fib[2]	2	00491DE8
fib[3]	3	00491DEC
fib[4]		00491DF0

```

int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```



fib	00491DE0	0012FF78
number	5	0012FF7C

fib[0]	1	00491DE0
fib[1]	1	00491DE4
fib[2]	2	00491DE8
fib[3]	3	00491DEC
fib[4]	5	00491DF0

```

int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```

fib	00491DE0	0012FF78
number	5	0012FF7C

```

int main()
{
    int number;
    cin >> number;
    int *fib;
    fib = new int[ number ];
    fib[ 0 ] = 1;
    fib[ 1 ] = 1;
    for( int i = 2; i < number; i++ )
        fib[ i ] = fib[ i - 2 ] + fib[ i - 1 ];
    for( int i = 0; i < number; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```

# Passing dynamically allocated arrays to functions

```
int main()
{
    int n;
    cout << "Enter a positive number ( 2 - 46 ) : ";
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[i] << endl;
    delete [] fib;
}
```

```
void fibonacci( int n, int *f )
{
    f[0] = 1;
    f[1] = 1;
    for( int i = 2; i < n; i++ )
        f[i] = f[i-2] + f[i-1];
}
```

```
int main()
{
    int n;
    cout << "Enter a positive number ( 2 - 46 ): ";
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

i		0012FF74
fib		0012FF78
n	5	0012FF7C

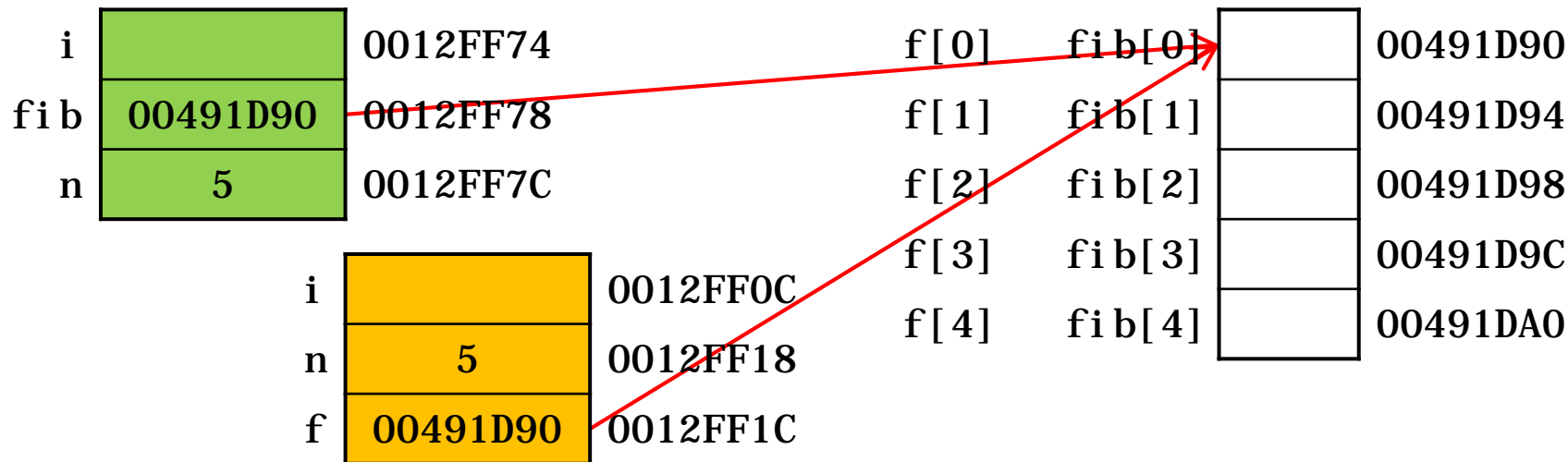
```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```



```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

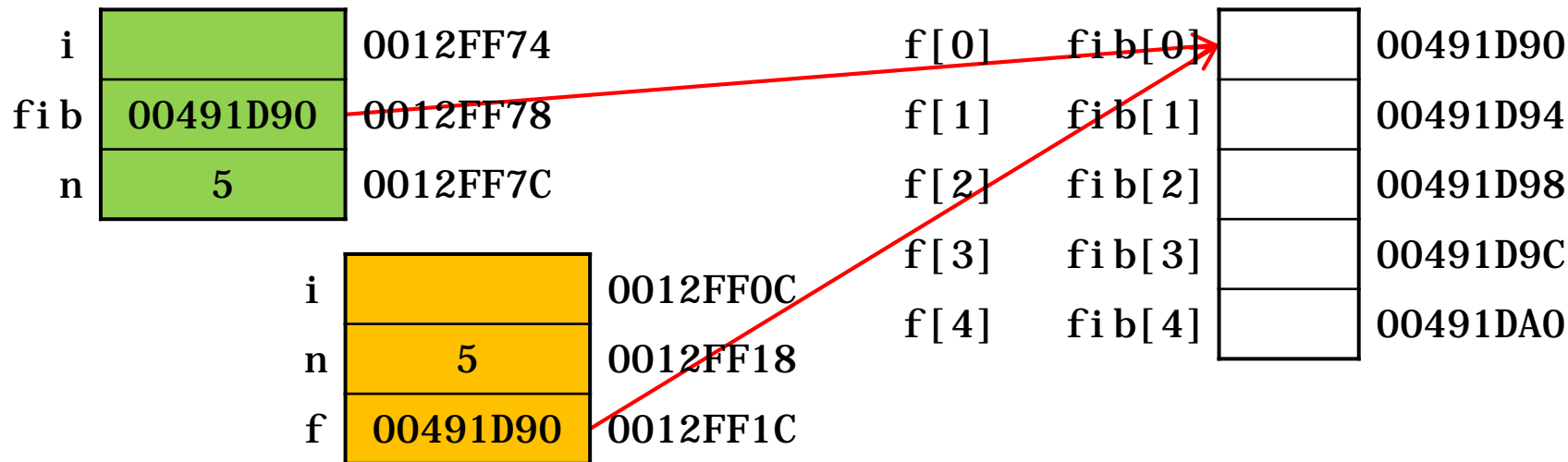
```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```



```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

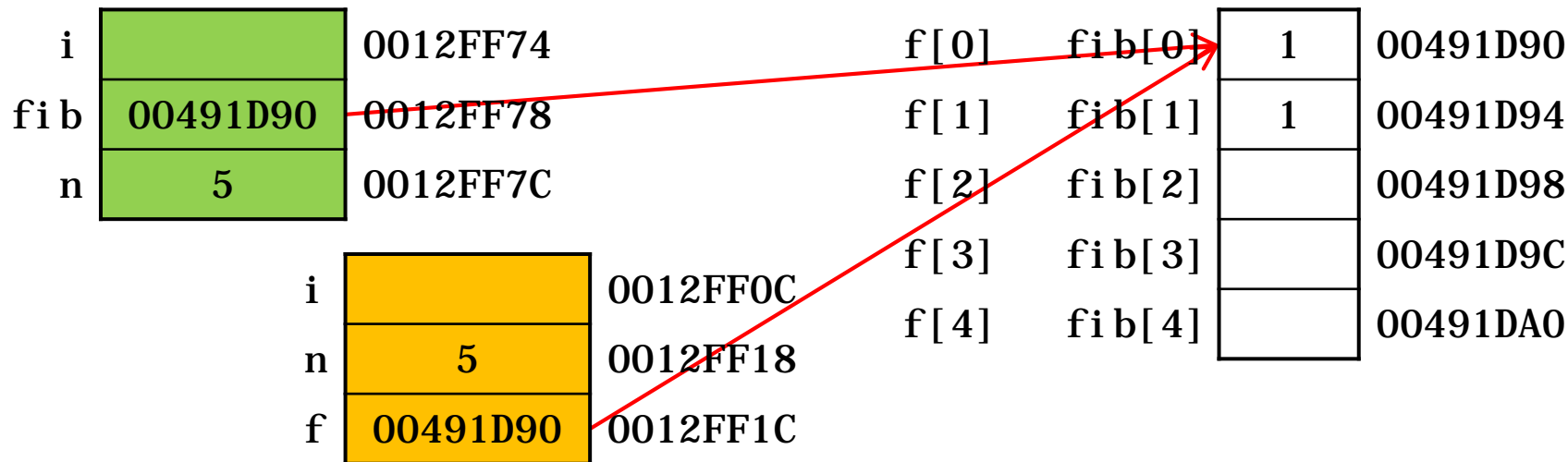
```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```





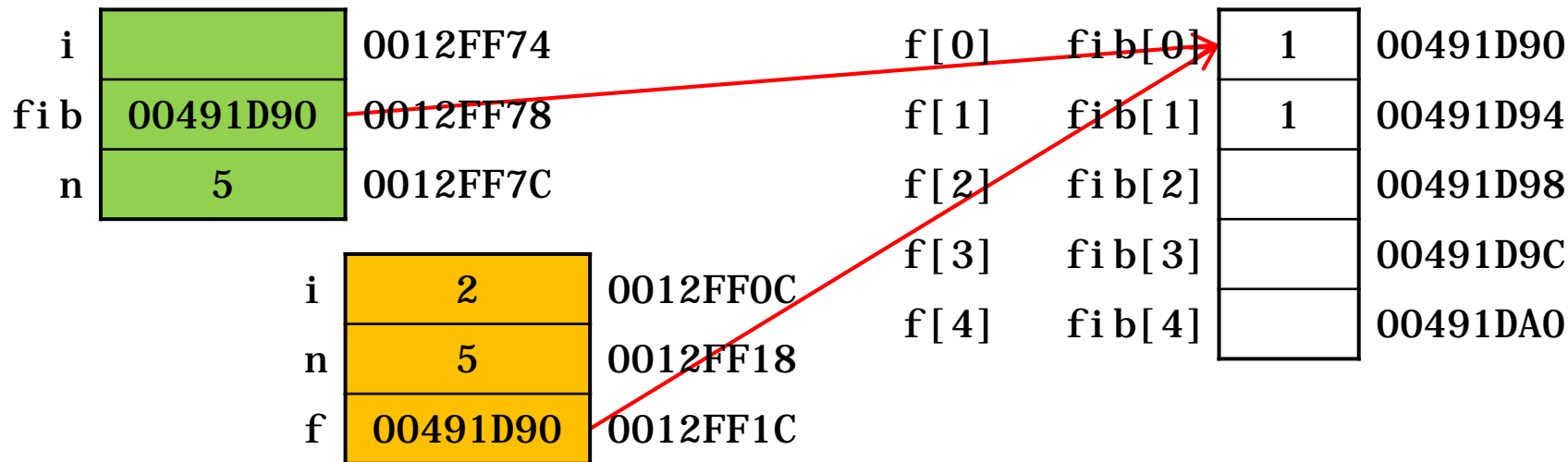
```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



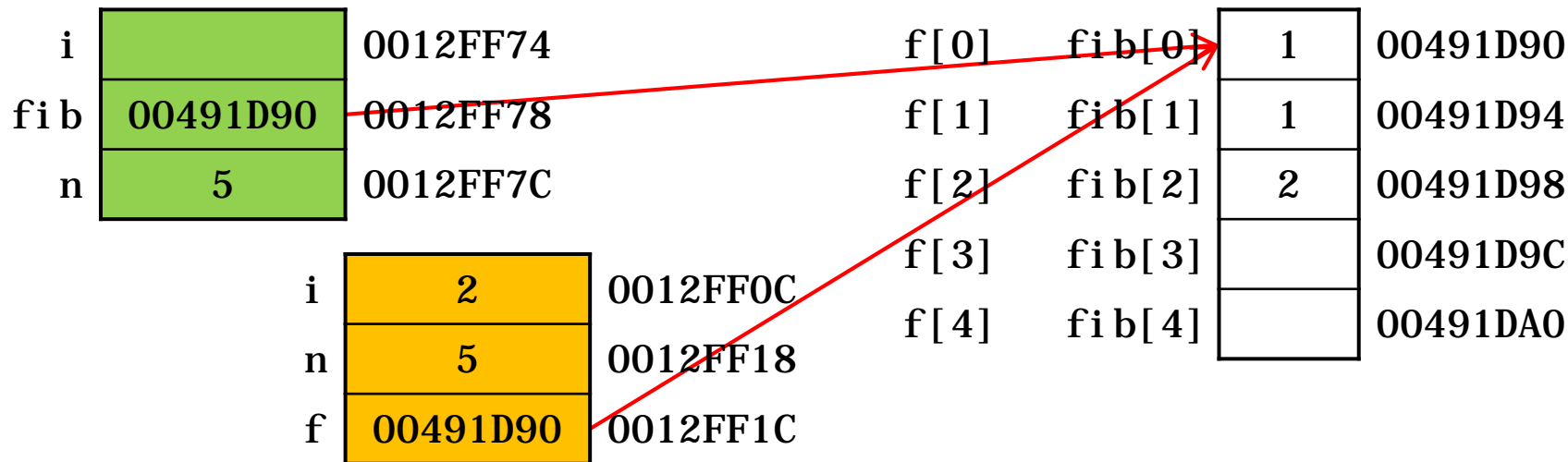
```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



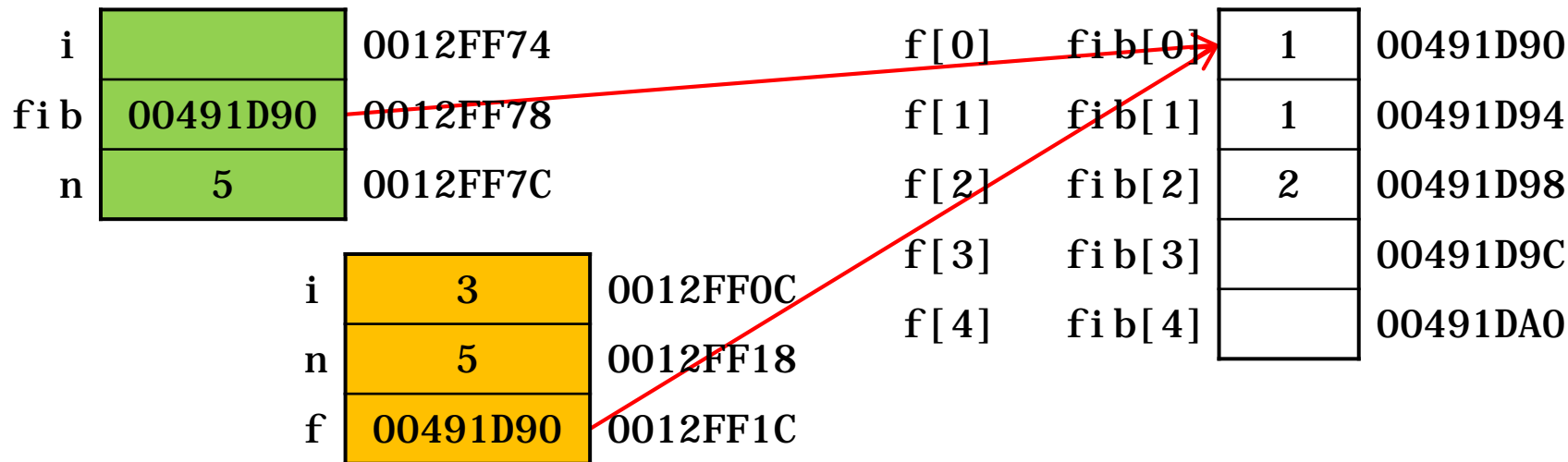
```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



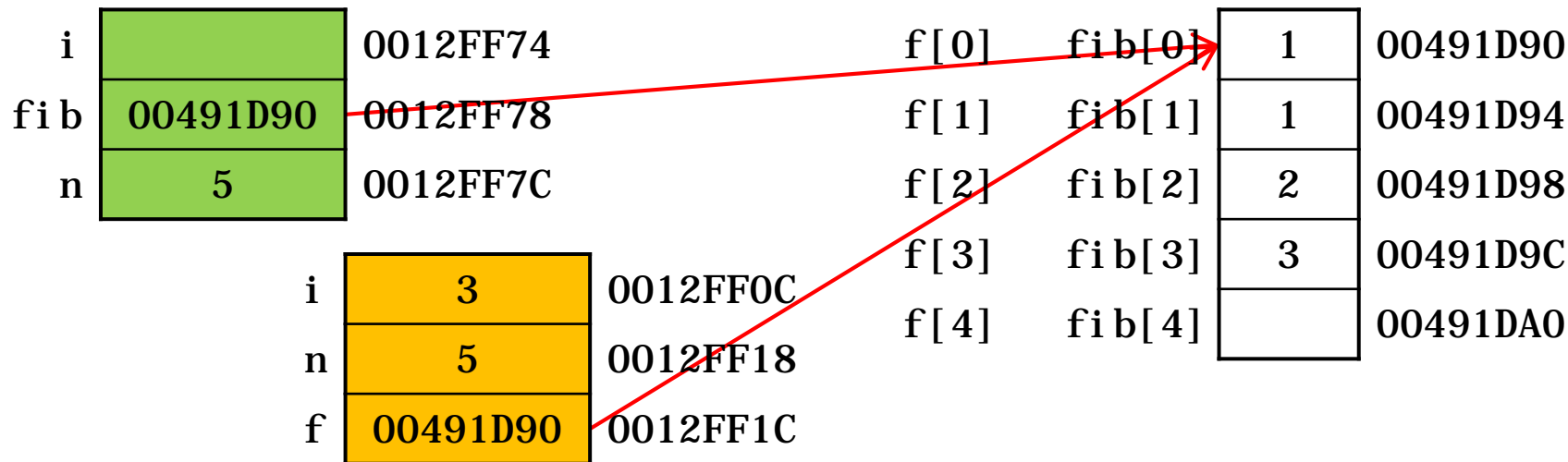
```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



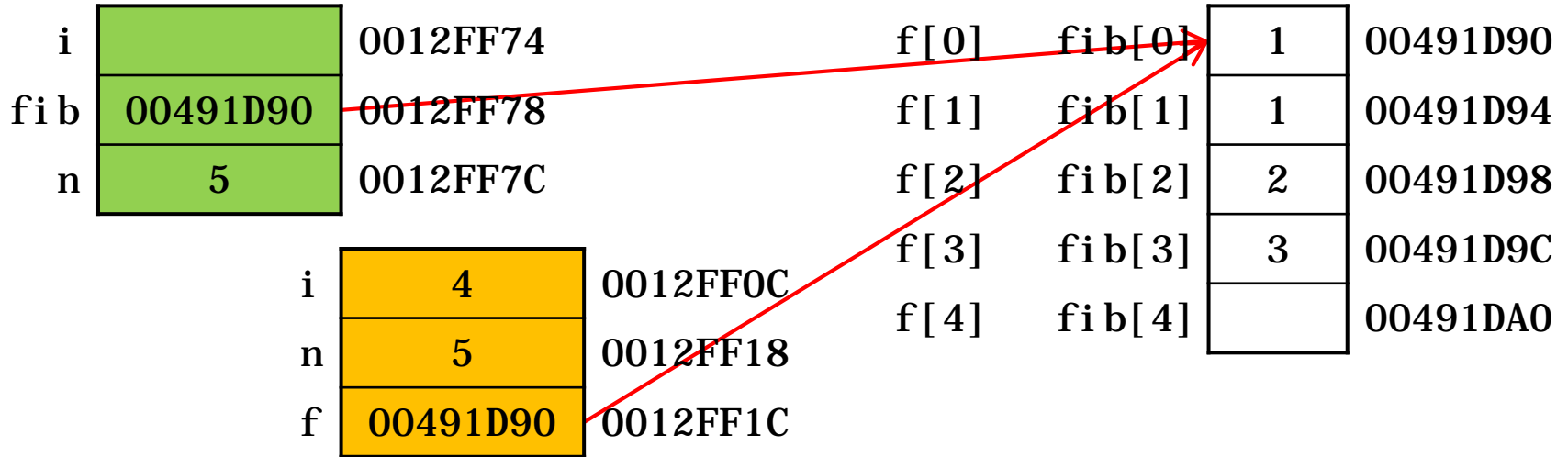
```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



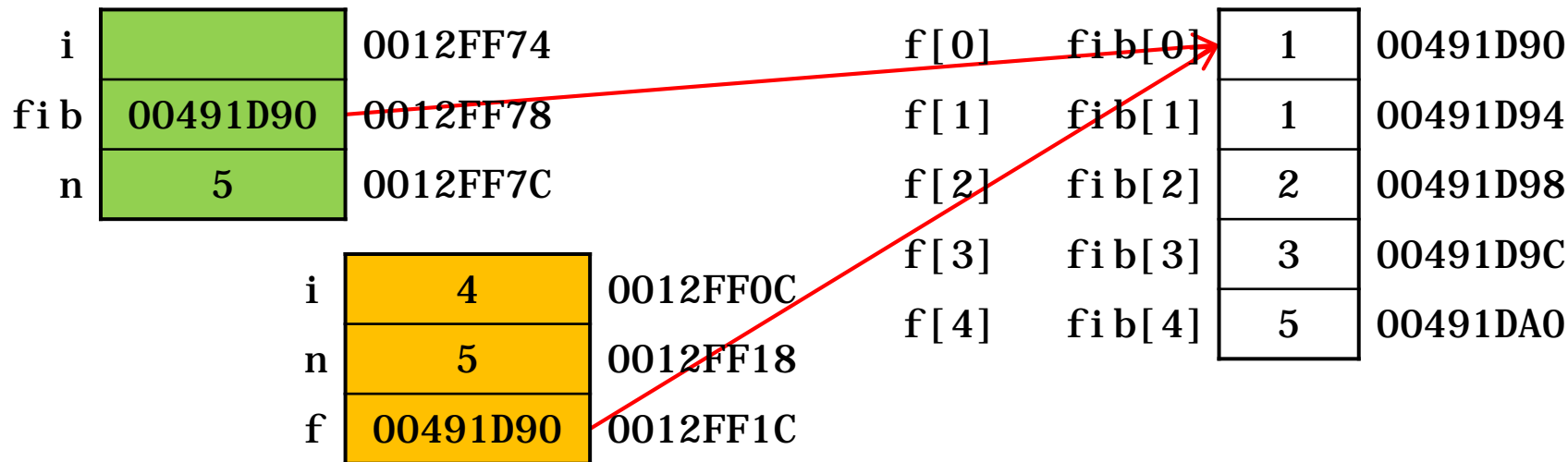
```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```





```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

i		0012FF74
fib	00491D90	0012FF78
n	5	0012FF7C

```
int main()
{
    int n;
    cin >> n;
    int *fib = new int[ n ];
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

```
void fibonacci( int n, int *f )
{
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

Returns dynamically allocated  
arrays back to the calling function

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[i] << endl;
    delete [] fib;
}
```

```
void fibonacci( int n, int *f )
{
    cout << "Enter a positive number ( 2 - 46 ): ";
    cin >> n;
    f = new int[ n ]
    f[0] = 1;
    f[1] = 1;
    for( int i = 2; i < n; i++ )
        f[i] = f[i-2] + f[i-1];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

void fibonacci( int n, int *f )
{
    cout << "Enter a positive number ( 2 - 46 ): ";
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

fib	0	0012FF78
n	0	0012FF7C

```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

fib	0	0012FF78
n	0	0012FF7C

i		0012FF0C
n	0	0012FF18
f	0	0012FF1C

```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

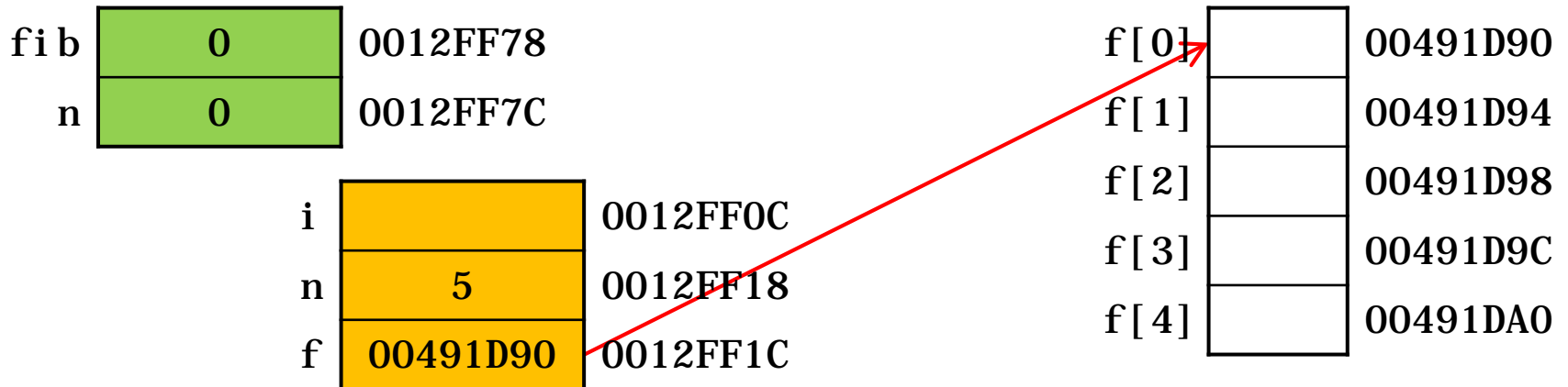
fib	0	0012FF78
n	0	0012FF7C

i		0012FF0C
n	5	0012FF18
f	0	0012FF1C

```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

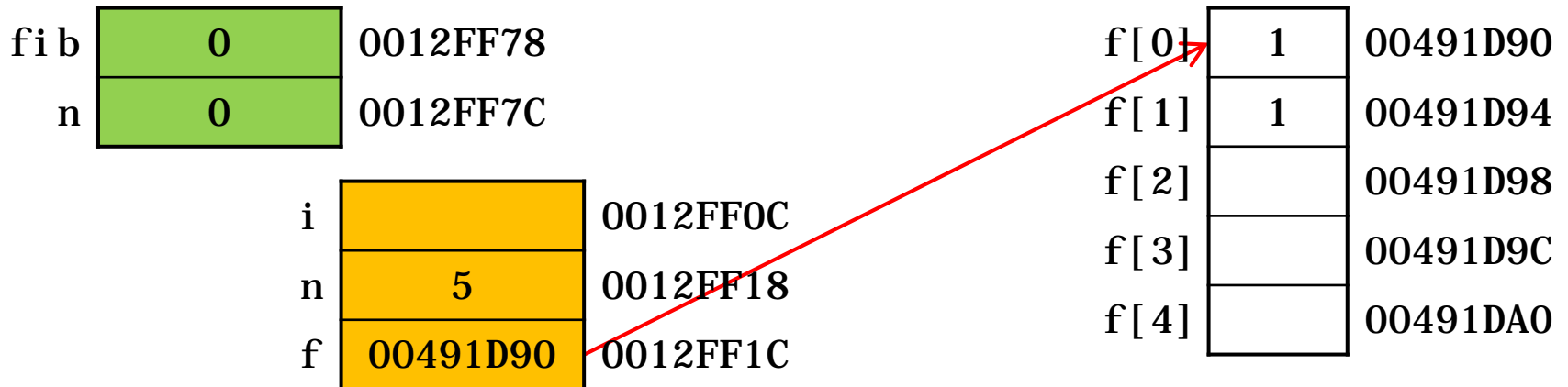
```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```





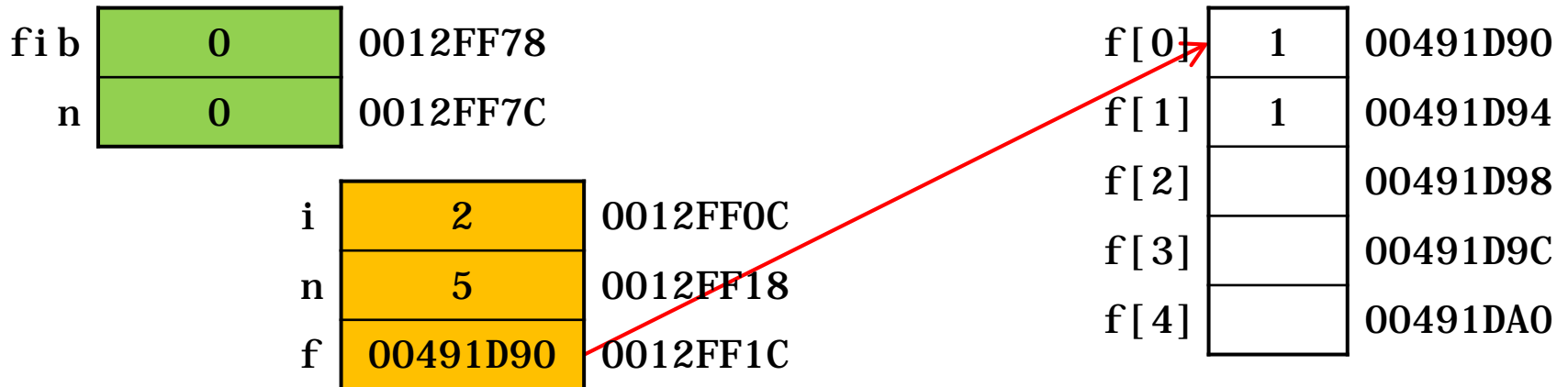
```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



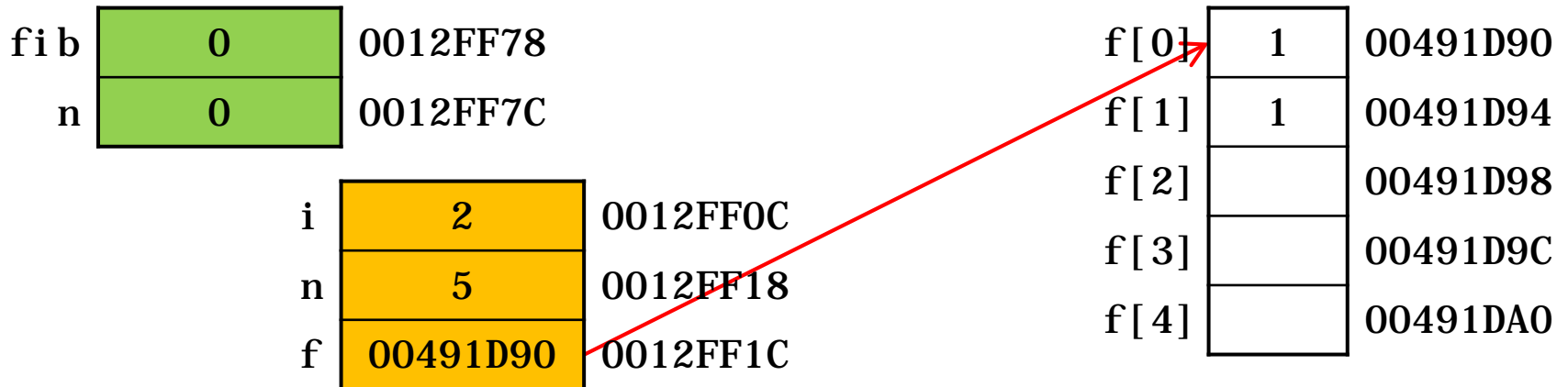
```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



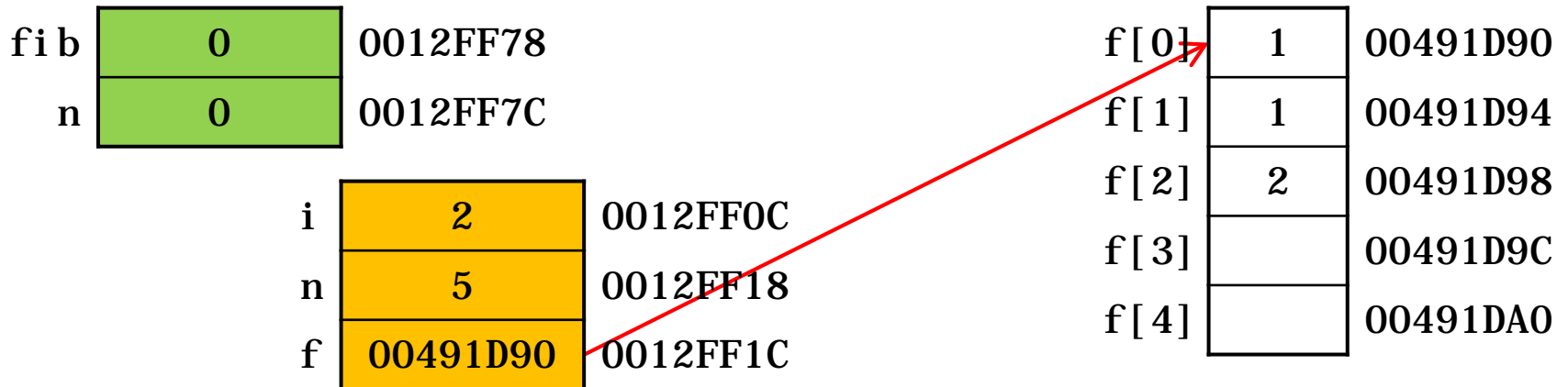
```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



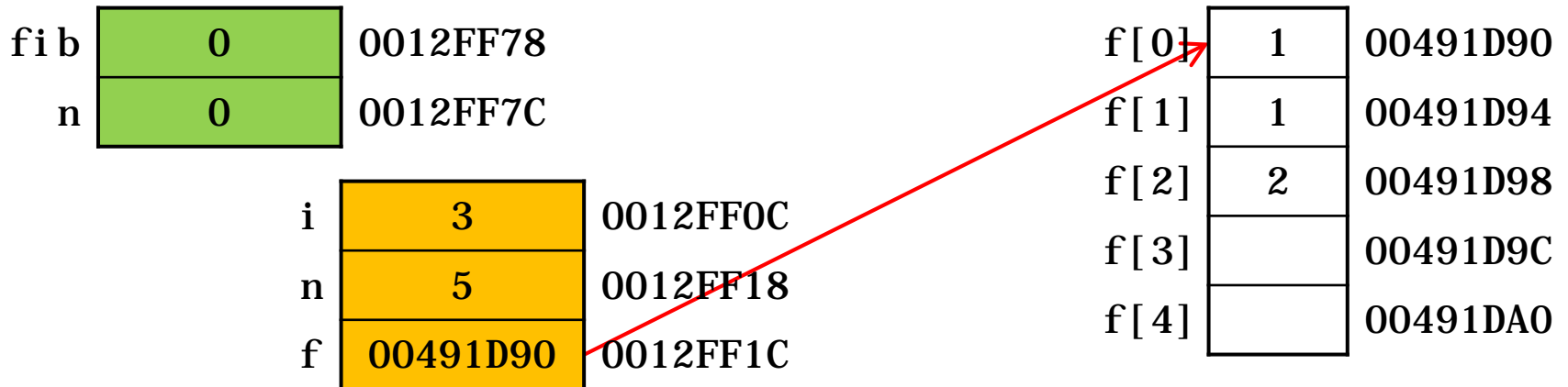
```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

fib	0	0012FF78	f[0]	1	00491D90
n	0	0012FF7C	f[1]	1	00491D94
			f[2]	2	00491D98
			f[3]	3	00491D9C
			f[4]		00491DA0

i	3	0012FF0C
n	5	0012FF18
f	00491D90	0012FF1C

```

void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}

```

```

int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```

fib	0	0012FF78	f[0]	1	00491D90
n	0	0012FF7C	f[1]	1	00491D94
			f[2]	2	00491D98
			f[3]	3	00491D9C
			f[4]		00491DA0

i	4	0012FF0C
n	5	0012FF18
f	00491D90	0012FF1C

```

void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}

```

```

int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

```



fib	0	0012FF78	f[0]	1	00491D90
n	0	0012FF7C	f[1]	1	00491D94
			f[2]	2	00491D98
			f[3]	3	00491D9C
			f[4]	5	00491DA0

i	4	0012FF0C
n	5	0012FF18
f	00491D90	0012FF1C

```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

fib	0	0012FF78
n	0	0012FF7C

f[0]	1	00491D90
f[1]	1	00491D94
f[2]	2	00491D98
f[3]	3	00491D9C
f[4]	5	00491DA0

```
void fibonacci( int n, int *f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int n = 0;
    int *fib = nullptr;
    fibonacci( n, fib );
    for( int i = 0; i < n; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}

void fibonacci( int &n, int *&f )
{
    cout << "Enter a positive number ( 2 - 46 ): ";
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

fib	0	0012FF78
num	0	0012FF7C

```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

f	fib	0	0012FF78
n	num	0	0012FF7C

i		0012FF0C
---	--	----------

```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

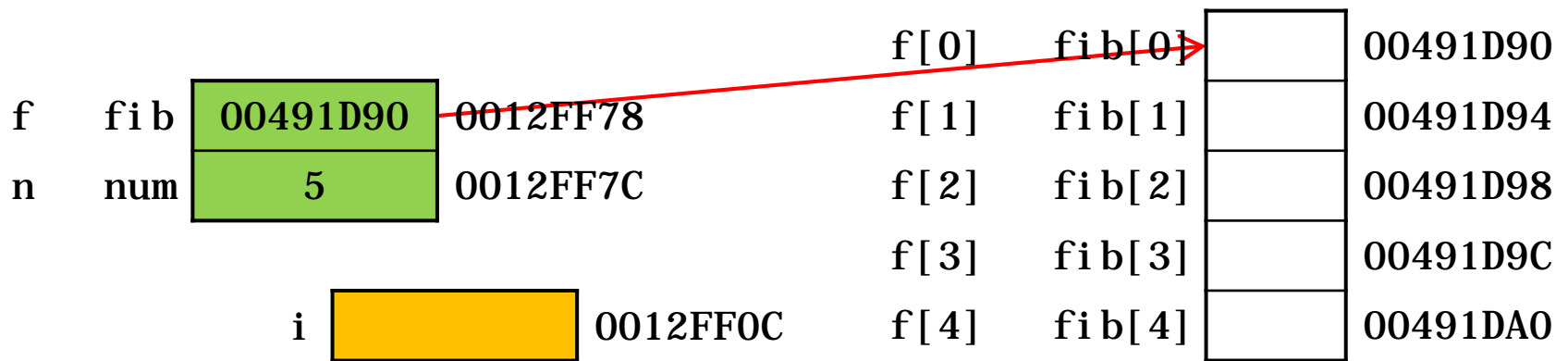
```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

f	fib	0	0012FF78
n	num	5	0012FF7C

i		0012FF0C
---	--	----------

```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

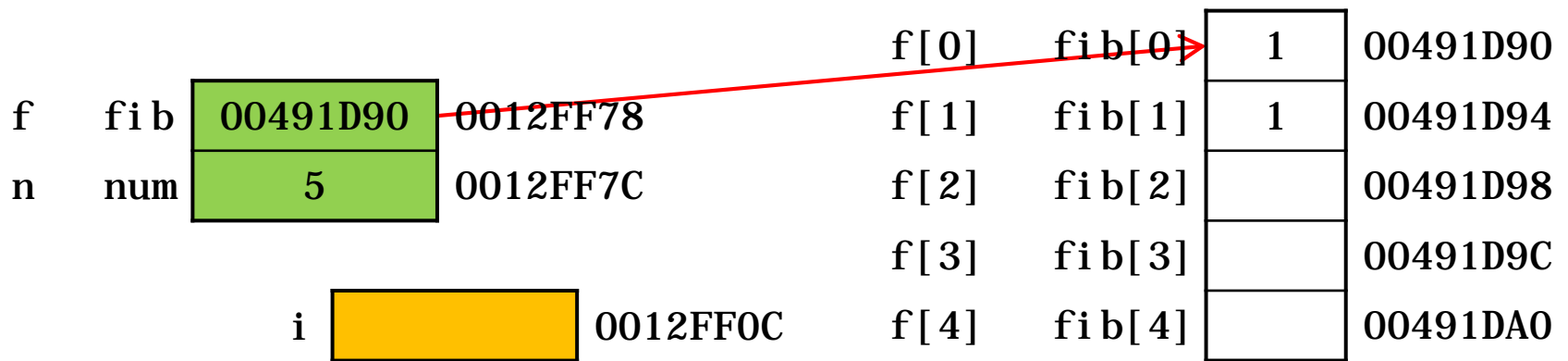
```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

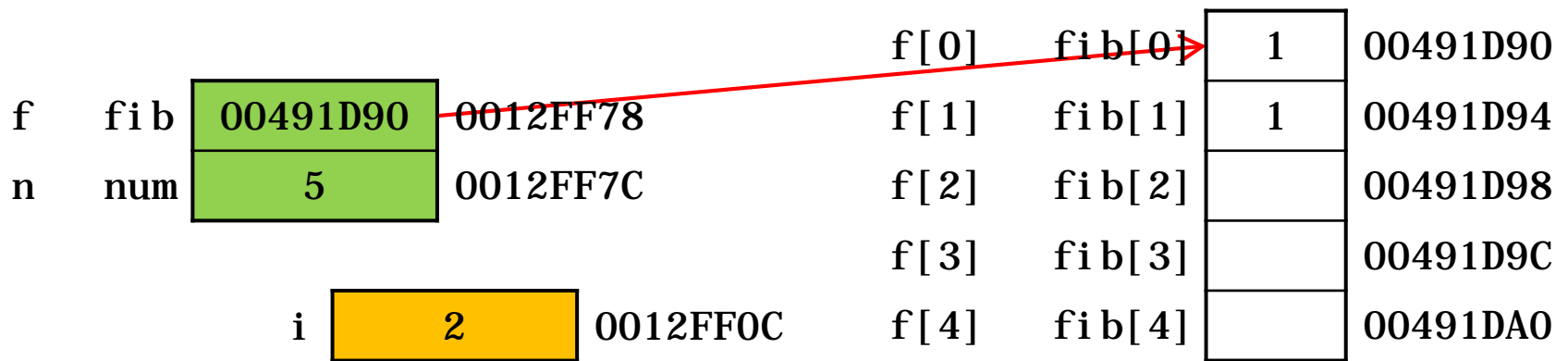
```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```





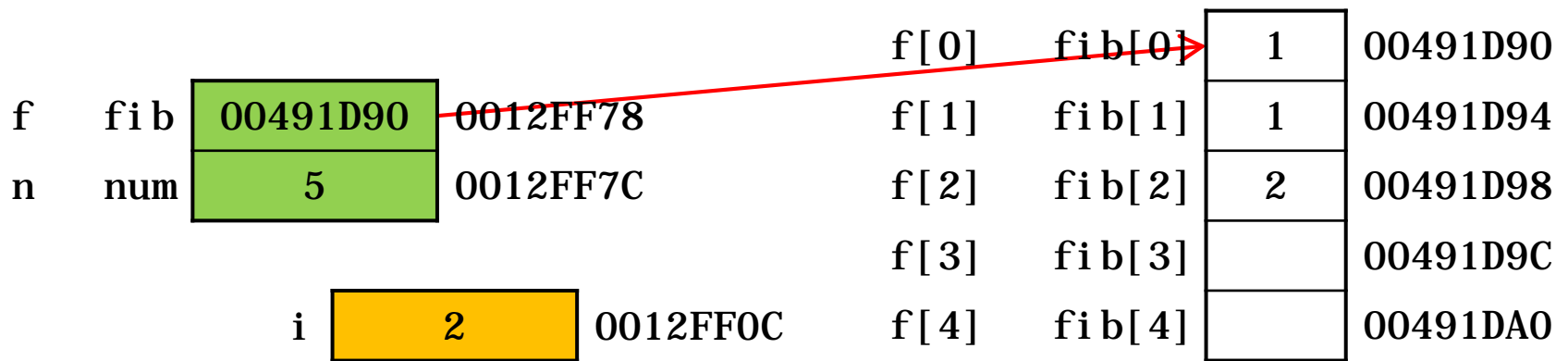
```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



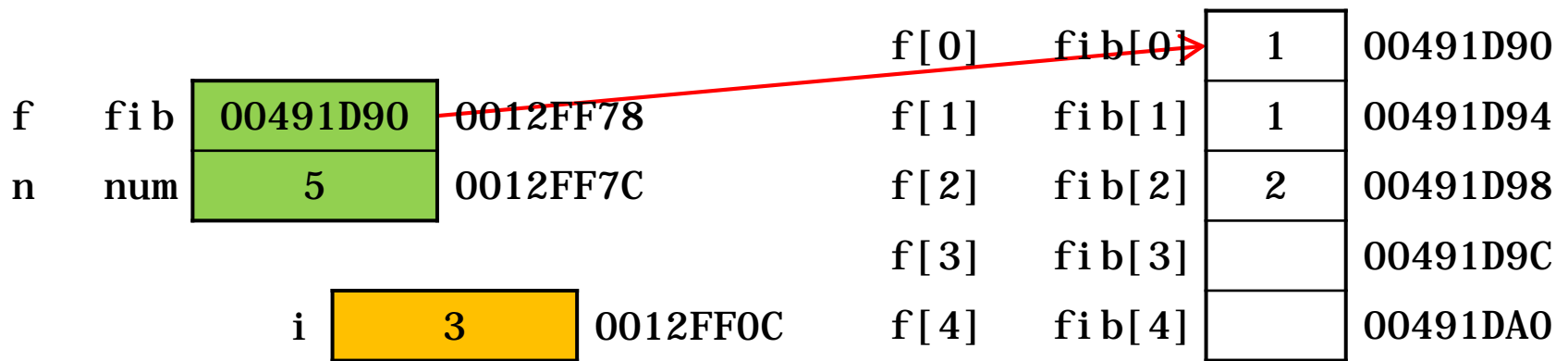
```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



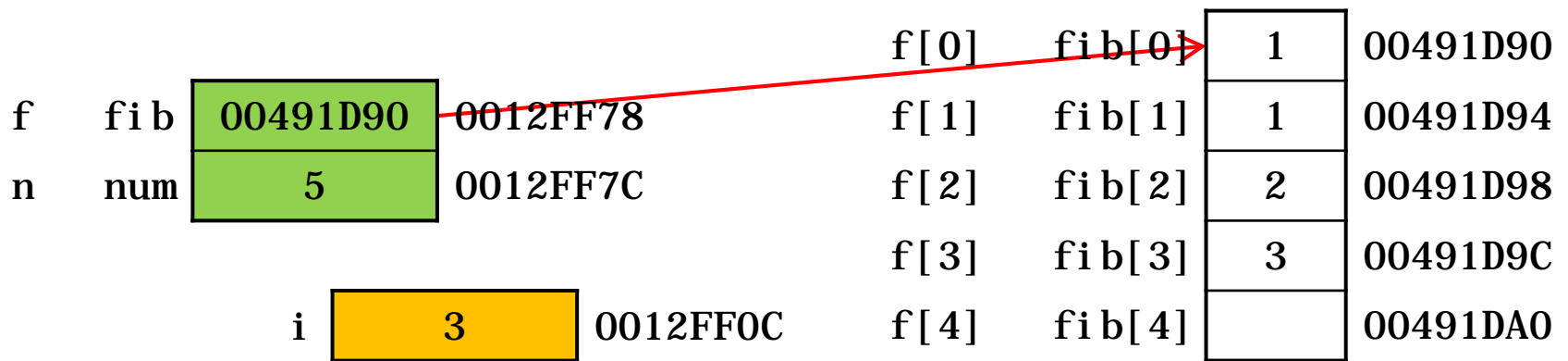
```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



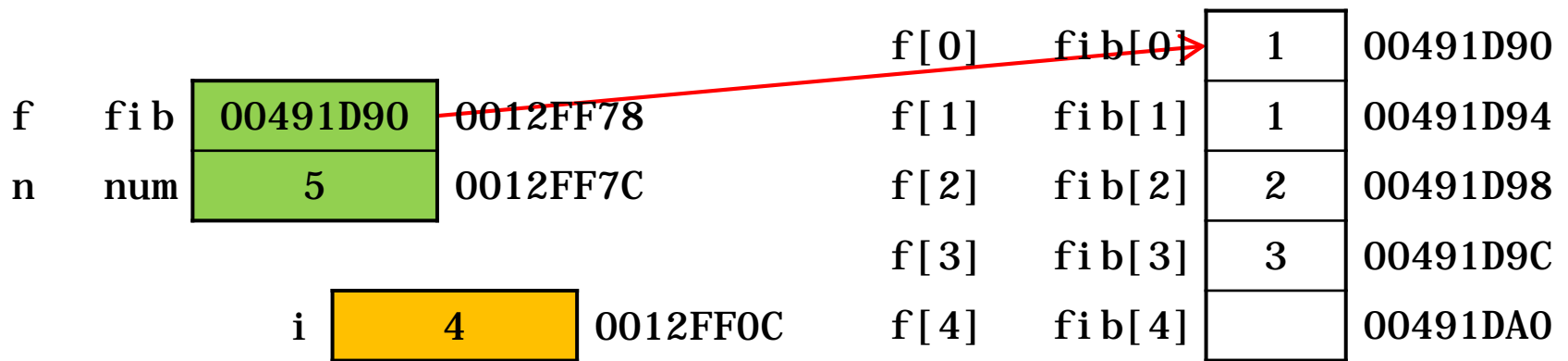
```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



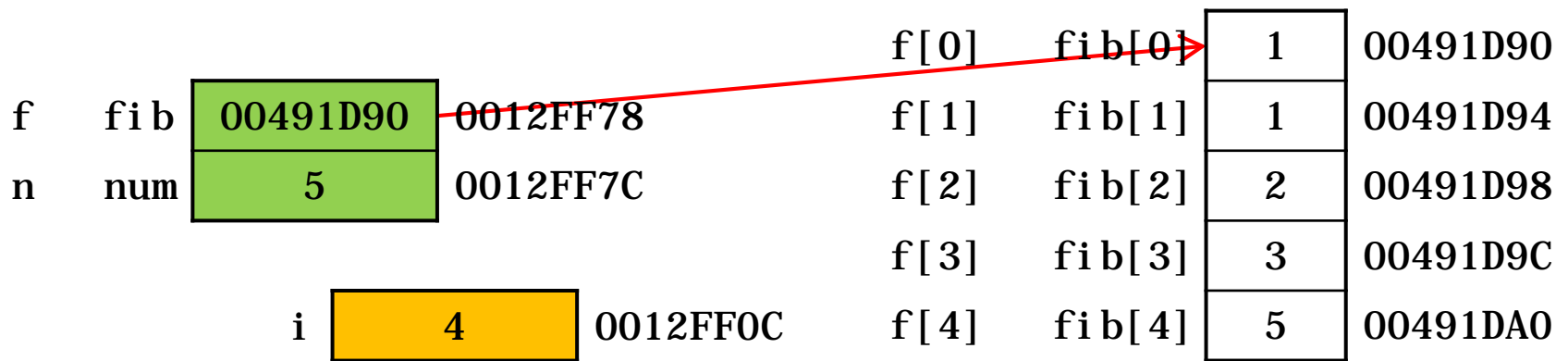
```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

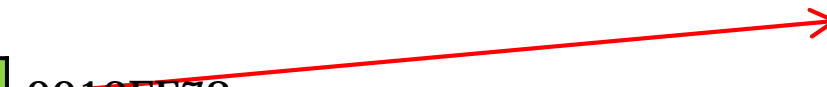


```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```



fib	00491D90	0012FF78
num	5	0012FF7C



```
void fibonacci( int &n, int *&f )
{
    cin >> n;
    f = new int[ n ];
    f[ 0 ] = 1;
    f[ 1 ] = 1;
    for( int i = 2; i < n; i++ )
        f[ i ] = f[ i - 2 ] + f[ i - 1 ];
}
```

```
int main()
{
    int num = 0;
    int *fib = nullptr;
    fibonacci( num, fib );
    for( int i = 0; i < num; i++ )
        cout << fib[ i ] << endl;
    delete[] fib;
}
```

Dynamically allocate a  
two dimensional array

```
int main()
{
    const int n = 4;
    int m;
    cout << "Enter a positive integer: ";
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    a[ 0 ][ 2 ] = 2;

    cout << "a[ 0 ][ 2 ] = " << a[ 0 ][ 2 ] << endl;

    delete[] a;
}
```

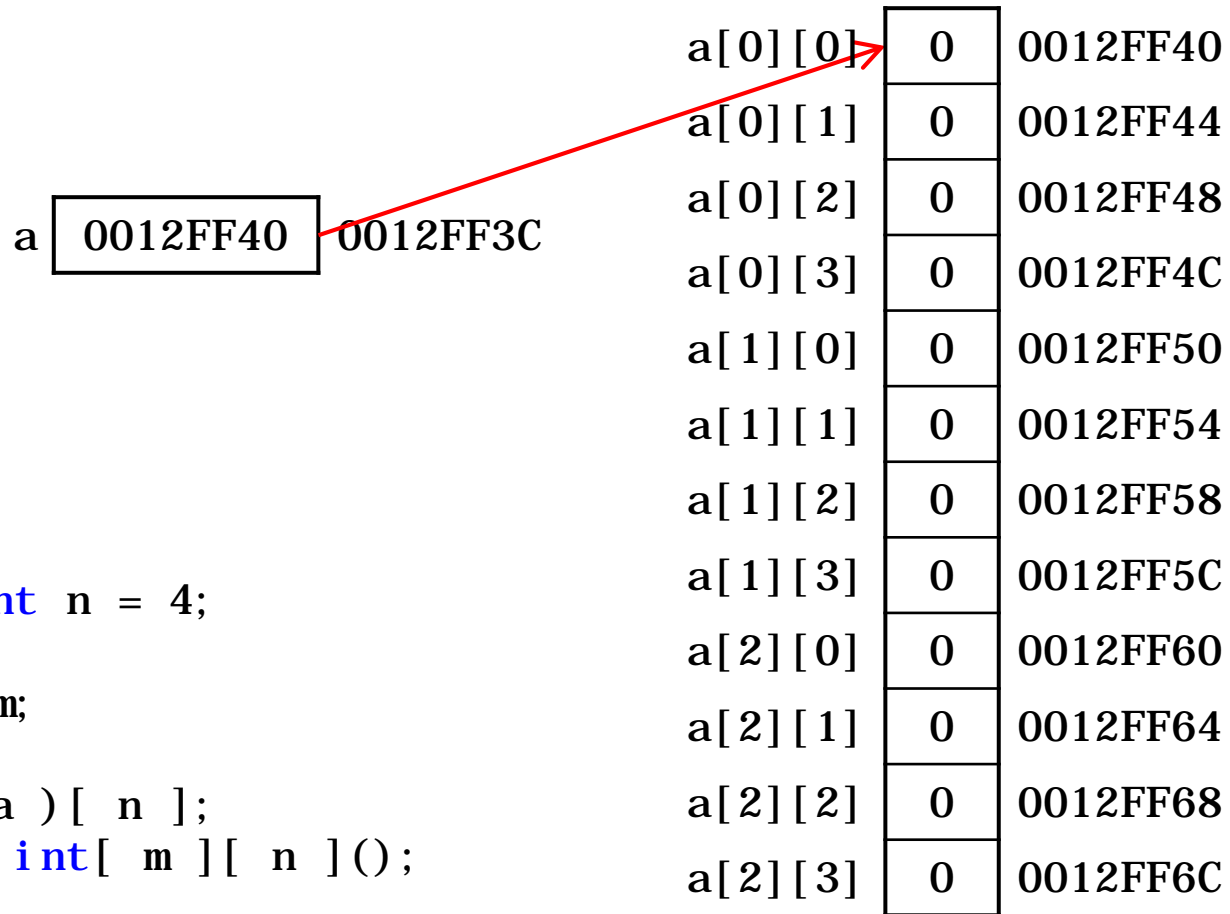
a  0012FF3C

```
int main()
{
    const int n = 4;
    int m;
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    a[ 0 ][ 2 ] = 2;

    delete[] a;
}
```

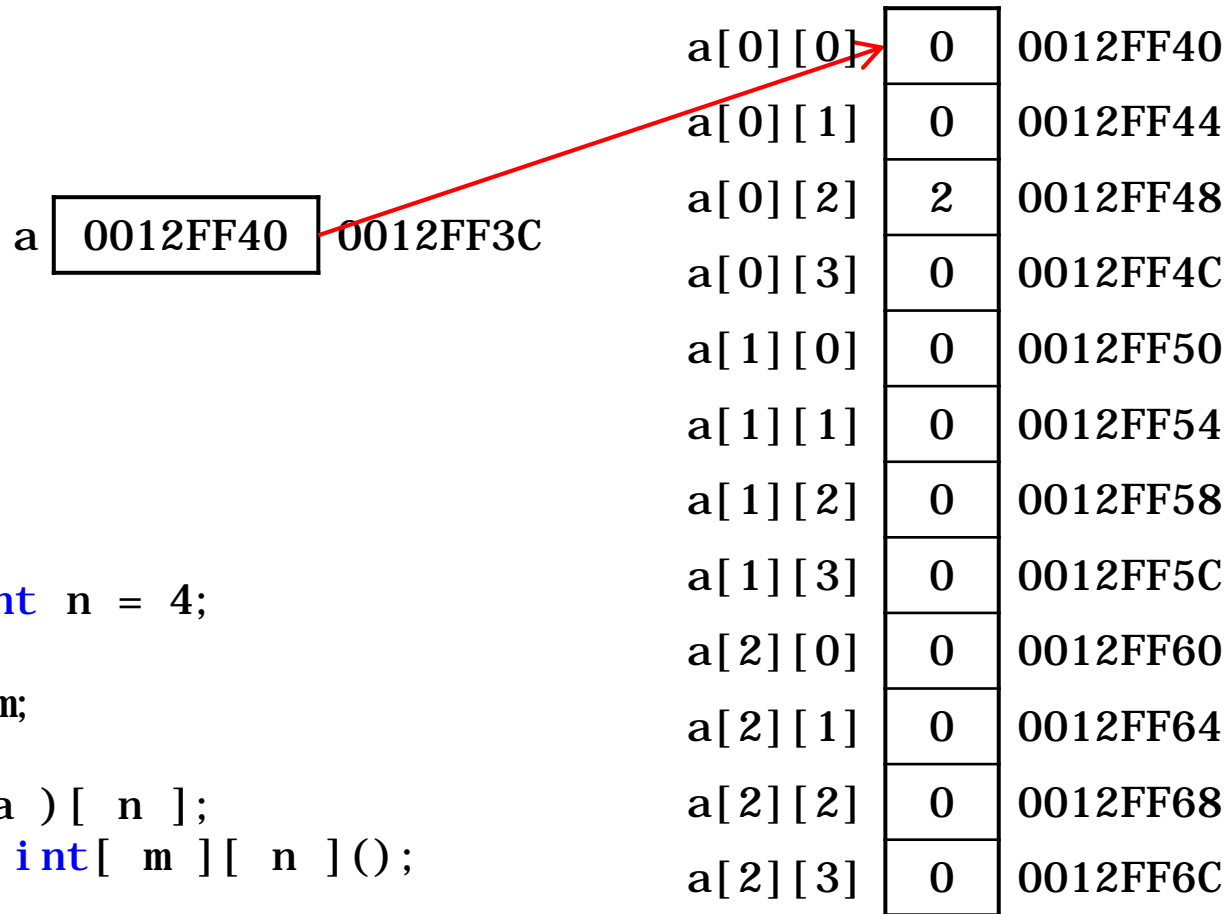


```
int main()
{
    const int n = 4;
    int m;
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    a[ 0 ][ 2 ] = 2;

    delete[] a;
}
```



```
int main()
{
    const int n = 4;
    int m;
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    a[ 0 ][ 2 ] = 2;

    delete[] a;
}
```

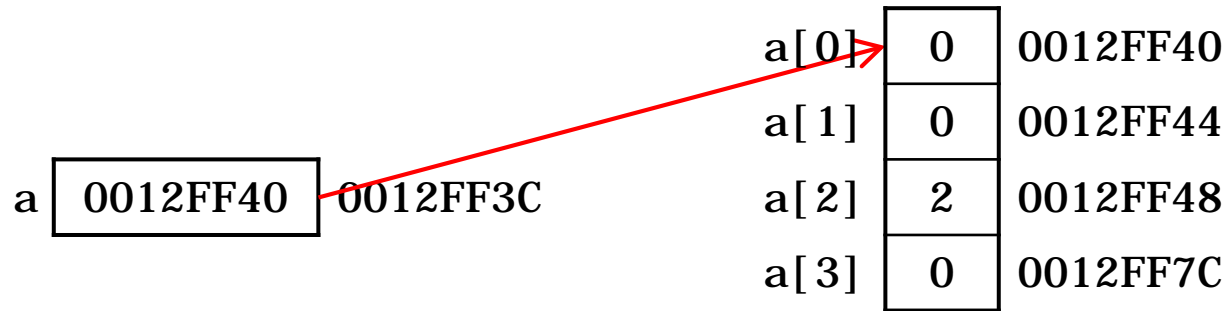
a 0012FF40 0012FF3C

```
int main()
{
    const int n = 4;
    int m;
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    a[ 0 ][ 2 ] = 2;

    delete[] a;
}
```

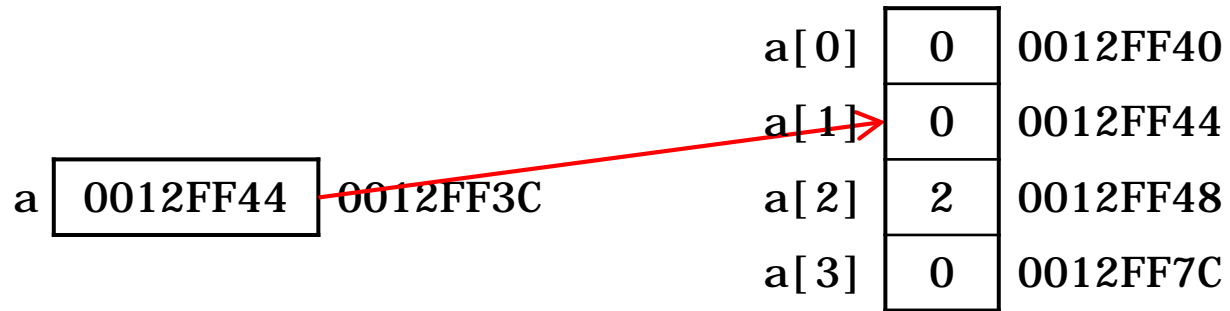


```
int main()
{
    int *a;
    a = new int[ 4 ];

    a++;

    delete [] a;
}
```

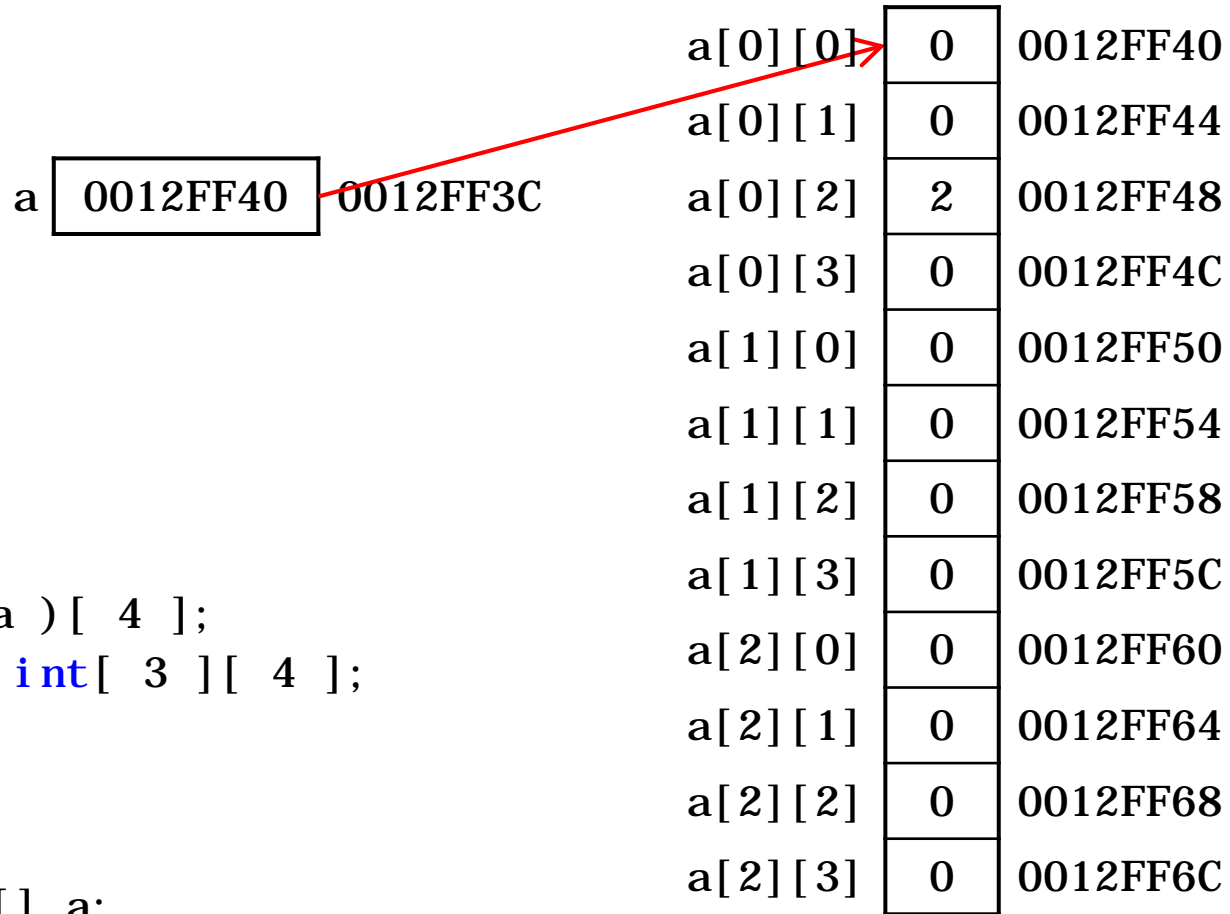




```
int main()
{
    int *a;
    a = new int[ 4 ];

    a++;

    delete [] a;
}
```



```
int main()
{
    int ( *a )[ 4 ];
    a = new int[ 3 ][ 4 ];

    a++;

    delete [] a;
}
```

a 

0012FF50
----------

 0012FF3C

a[0][0]	0	0012FF40
a[0][1]	0	0012FF44
a[0][2]	2	0012FF48
a[0][3]	0	0012FF4C
a[1][0]	0	0012FF50
a[1][1]	0	0012FF54
a[1][2]	0	0012FF58
a[1][3]	0	0012FF5C
a[2][0]	0	0012FF60
a[2][1]	0	0012FF64
a[2][2]	0	0012FF68
a[2][3]	0	0012FF6C

```
int main()
{
    int ( *a )[ 4 ];
    a = new int[ 3 ][ 4 ];

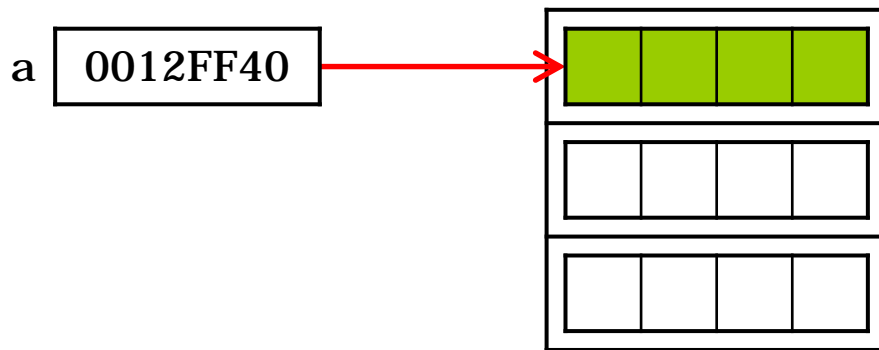
    a++;

    delete [] a;
}
```



```
int *a;  
a = new int[ 4 ];  
a++;
```

## Pointer



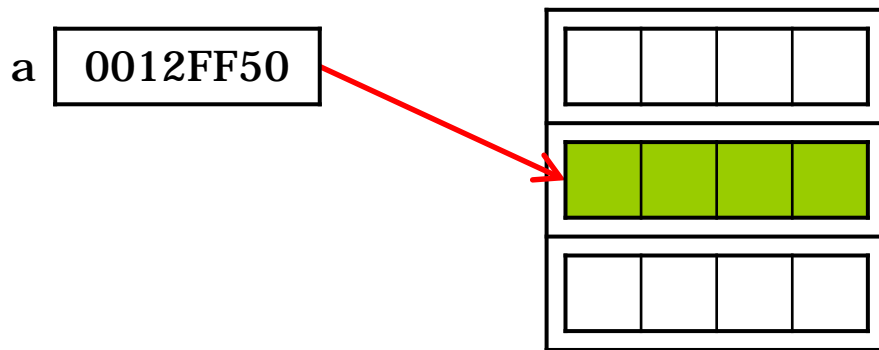
```
int ( *a )[ 4 ];  
a = new int[ 3 ][ 4 ];  
a++;
```

## Pointer to array



```
int *a;  
a = new int[ 4 ];  
a++;
```

## Pointer



```
int ( *a )[ 4 ];  
a = new int[ 3 ][ 4 ];  
a++;
```

## Pointer to array

Passing dynamically allocated  
two dimensional arrays to functions

```

const int n = 4;
void fun( int ( *p )[ n ] );

int main()
{
    int m;
    cout << "Enter a positive integer: ";
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    fun( a );

    cout << "a[ 0 ][ 2 ] = " << a[ 0 ][ 2 ] << endl;

    delete[] a;
}

void fun( int ( *p )[ n ] )
{
    p[ 0 ][ 2 ] = 2;
}

```

a  0012FF3C

```
const int n = 4;
void fun( int ( *p )[ n ] );

int main()
{
    int m;
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    fun( a );

    delete[] a;
}

void fun( int ( *p )[ n ] )
{
    p[ 0 ][ 2 ] = 2;
}
```



a  0012FF3C

```
const int n = 4;
void fun( int ( *p )[ n ] );

int main()
{
    int m;
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    fun( a );

    delete[] a;
}

void fun( int ( *p )[ n ] )
{
    p[ 0 ][ 2 ] = 2;
}
```

a 0012FF40 0012FF3C

```
const int n = 4;
void fun( int ( *p ) [ n ] );

int main()
{
    int m;
    cin >> m;

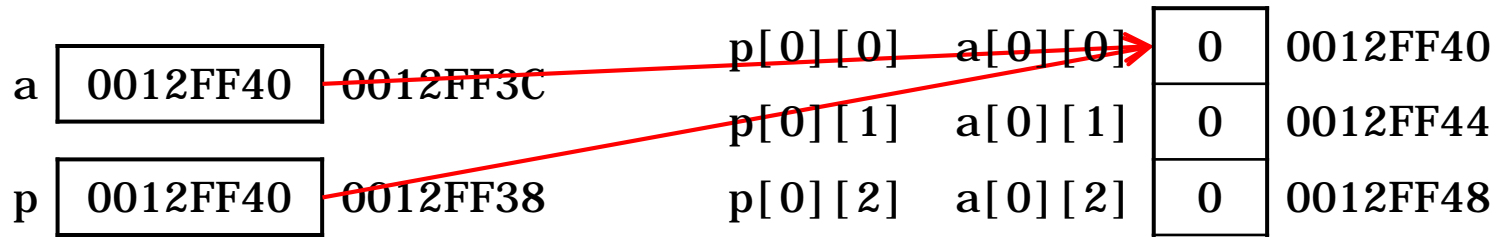
    int ( *a ) [ n ];
    a = new int [ m ] [ n ]();

    fun( a );

    delete[] a;
}

void fun( int ( *p ) [ n ] )
{
    p[ 0 ] [ 2 ] = 2;
}
```

a[0][0]	0	0012FF40
a[0][1]	0	0012FF44
a[0][2]	0	0012FF48
a[0][3]	0	0012FF4C
a[1][0]	0	0012FF50
a[1][1]	0	0012FF54
a[1][2]	0	0012FF58
a[1][3]	0	0012FF5C
a[2][0]	0	0012FF60
a[2][1]	0	0012FF64
a[2][2]	0	0012FF68
a[2][3]	0	0012FF6C



```

const int n = 4;
void fun( int ( *p ) [ n ] );

int main()
{
    int m;
    cin >> m;

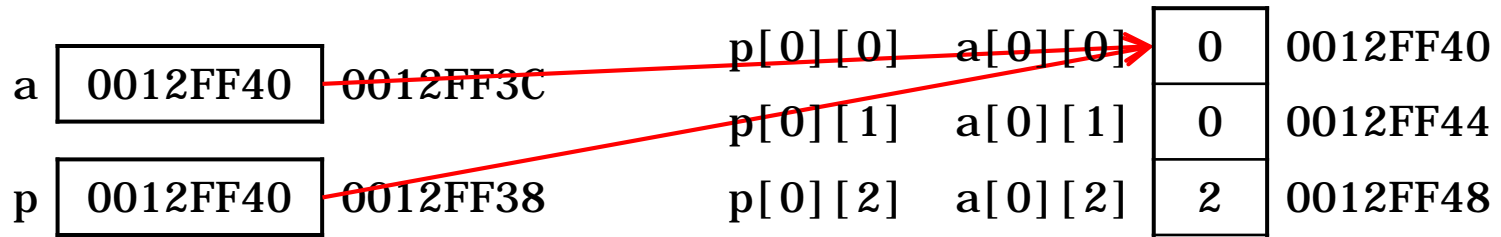
    int ( *a ) [ n ];
    a = new int [ m ] [ n ]();

    fun( a );

    delete[] a;
}

void fun( int ( *p ) [ n ] )
{
    p[ 0 ] [ 2 ] = 2;
}

```



```
const int n = 4;
void fun( int ( *p ) [ n ] );
```

```
int main()
{
    int m;
    cin >> m;

    int ( *a ) [ n ];
    a = new int [ m ] [ n ] ();

    fun( a );

    delete[] a;
}
```

```
void fun( int ( *p ) [ n ] )
{
    p[ 0 ] [ 2 ] = 2;
}
```

p[0][0]	a[0][0]	0	0012FF40
p[0][1]	a[0][1]	0	0012FF44
p[0][2]	a[0][2]	2	0012FF48
p[0][3]	a[0][3]	0	0012FF4C
p[1][0]	a[1][0]	0	0012FF50
p[1][1]	a[1][1]	0	0012FF54
p[1][2]	a[1][2]	0	0012FF58
p[1][3]	a[1][3]	0	0012FF5C
p[2][0]	a[2][0]	0	0012FF60
p[2][1]	a[2][1]	0	0012FF64
p[2][2]	a[2][2]	0	0012FF68
p[2][3]	a[2][3]	0	0012FF6C



```
const int n = 4;  
void fun( int ( *p ) [ n ] );
```

```
int main()  
{  
    int m;  
    cin >> m;  
  
    int ( *a ) [ n ];  
    a = new int [ m ] [ n ]();  
  
    fun( a );  
  
    delete[] a;  
}
```

```
void fun( int ( *p ) [ n ] )  
{  
    p[ 0 ] [ 2 ] = 2;  
}
```

a[0][0]	0	0012FF40
a[0][1]	0	0012FF44
a[0][2]	2	0012FF48
a[0][3]	0	0012FF4C
a[1][0]	0	0012FF50
a[1][1]	0	0012FF54
a[1][2]	0	0012FF58
a[1][3]	0	0012FF5C
a[2][0]	0	0012FF60
a[2][1]	0	0012FF64
a[2][2]	0	0012FF68
a[2][3]	0	0012FF6C

a 0012FF40 0012FF3C

```
const int n = 4;
void fun( int ( *p )[ n ] );

int main()
{
    int m;
    cin >> m;

    int ( *a )[ n ];
    a = new int[ m ][ n ]();

    fun( a );

    delete[] a;
}

void fun( int ( *p )[ n ] )
{
    p[ 0 ][ 2 ] = 2;
}
```

Dynamically allocate  
an array of arrays

```
int main()
{
    int n;
    cout << "Enter a positive integer: ";
    cin >> n;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];

    a[ 1 ][ 1 ] = 3;

    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```

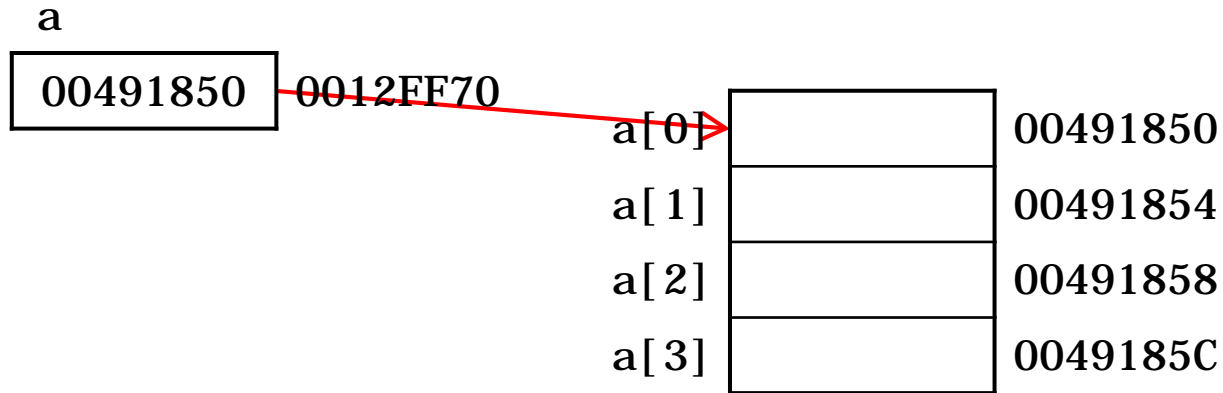


a

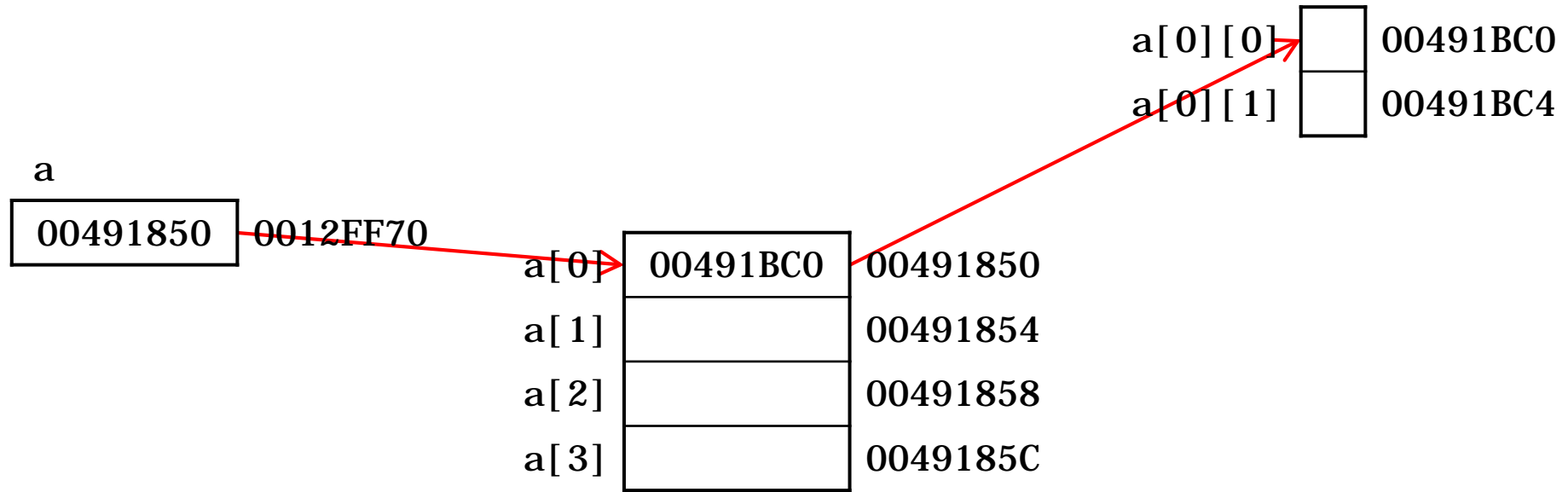


0012FF70

```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```



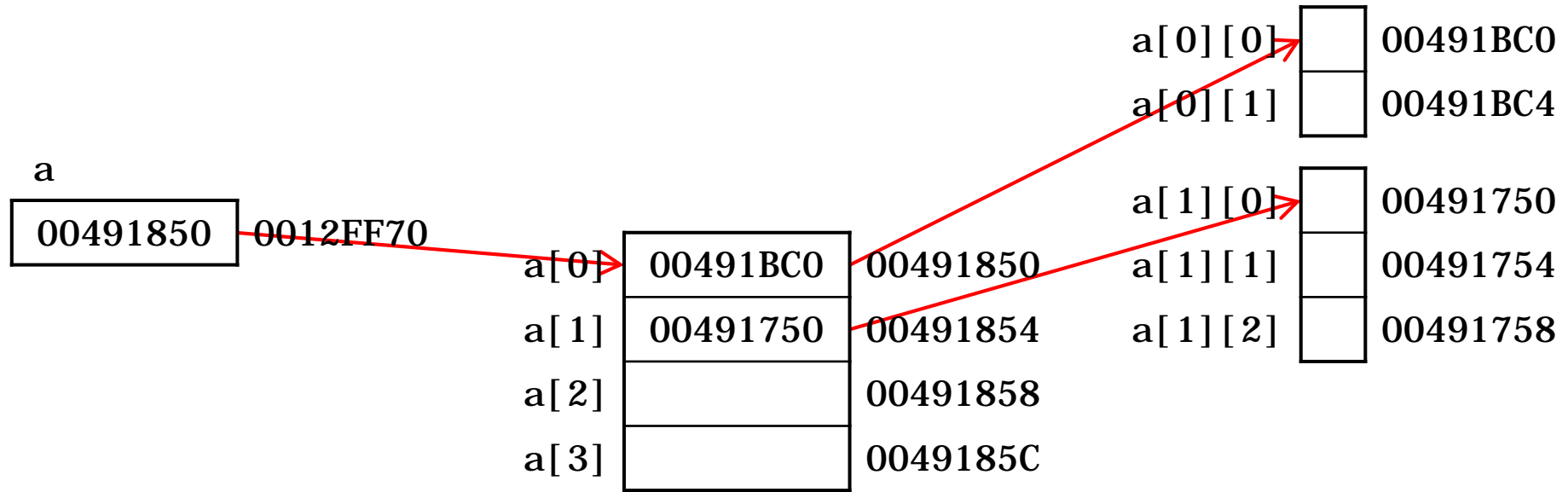
```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```



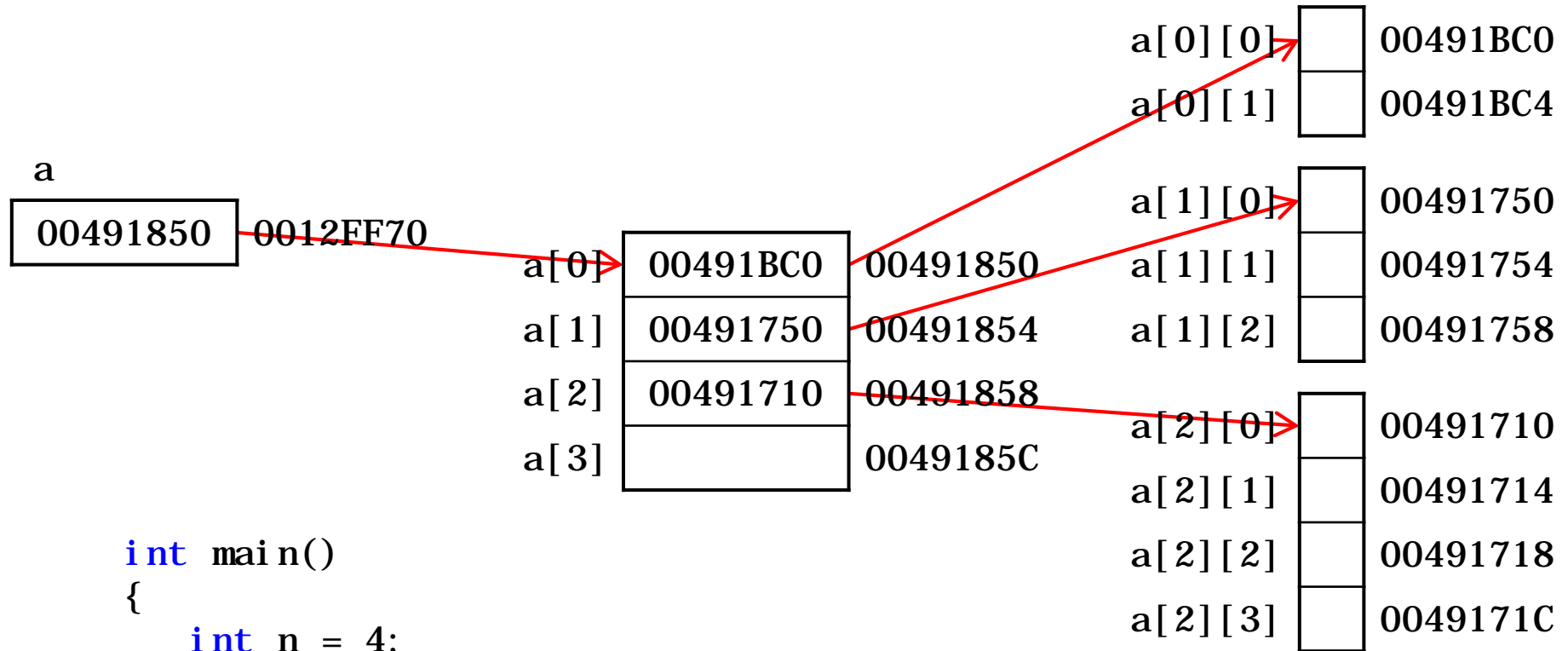
```

int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}

```



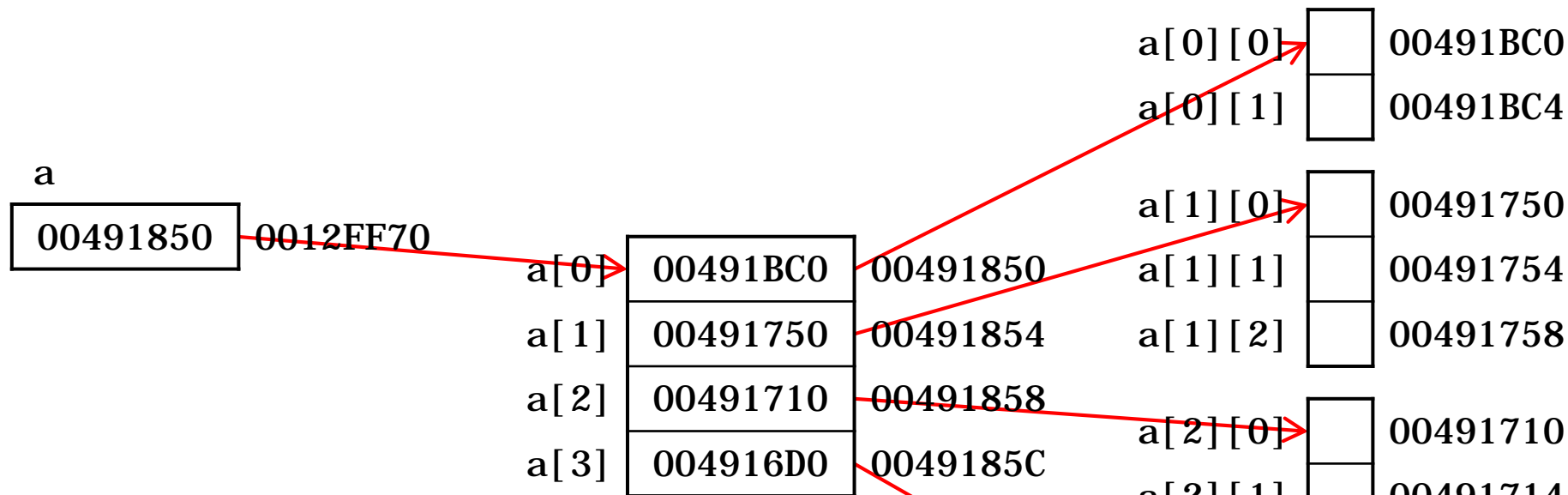
```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```



```

int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}

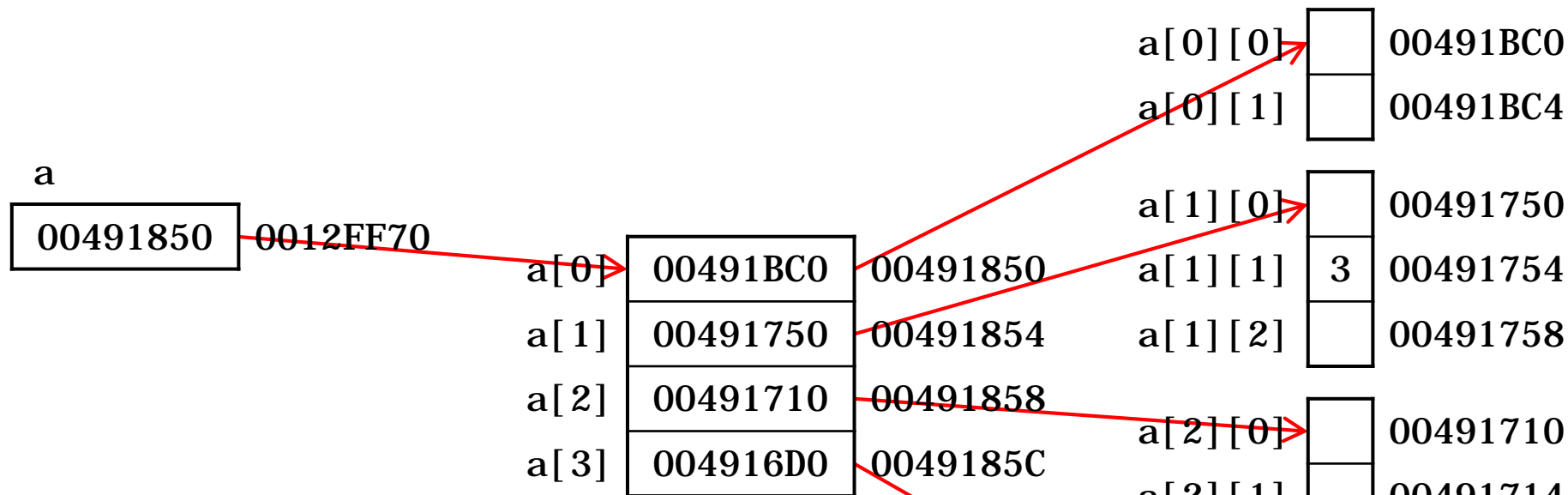
```



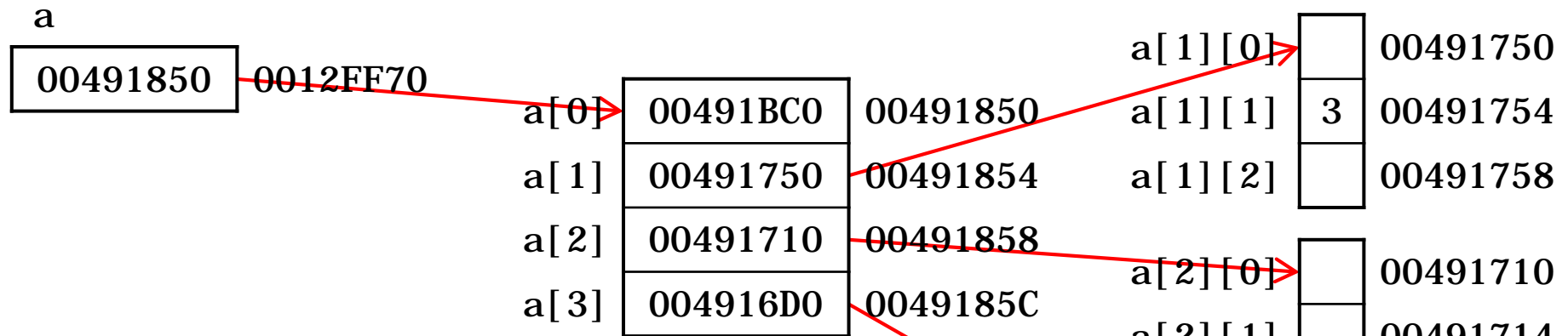
```

int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}

```

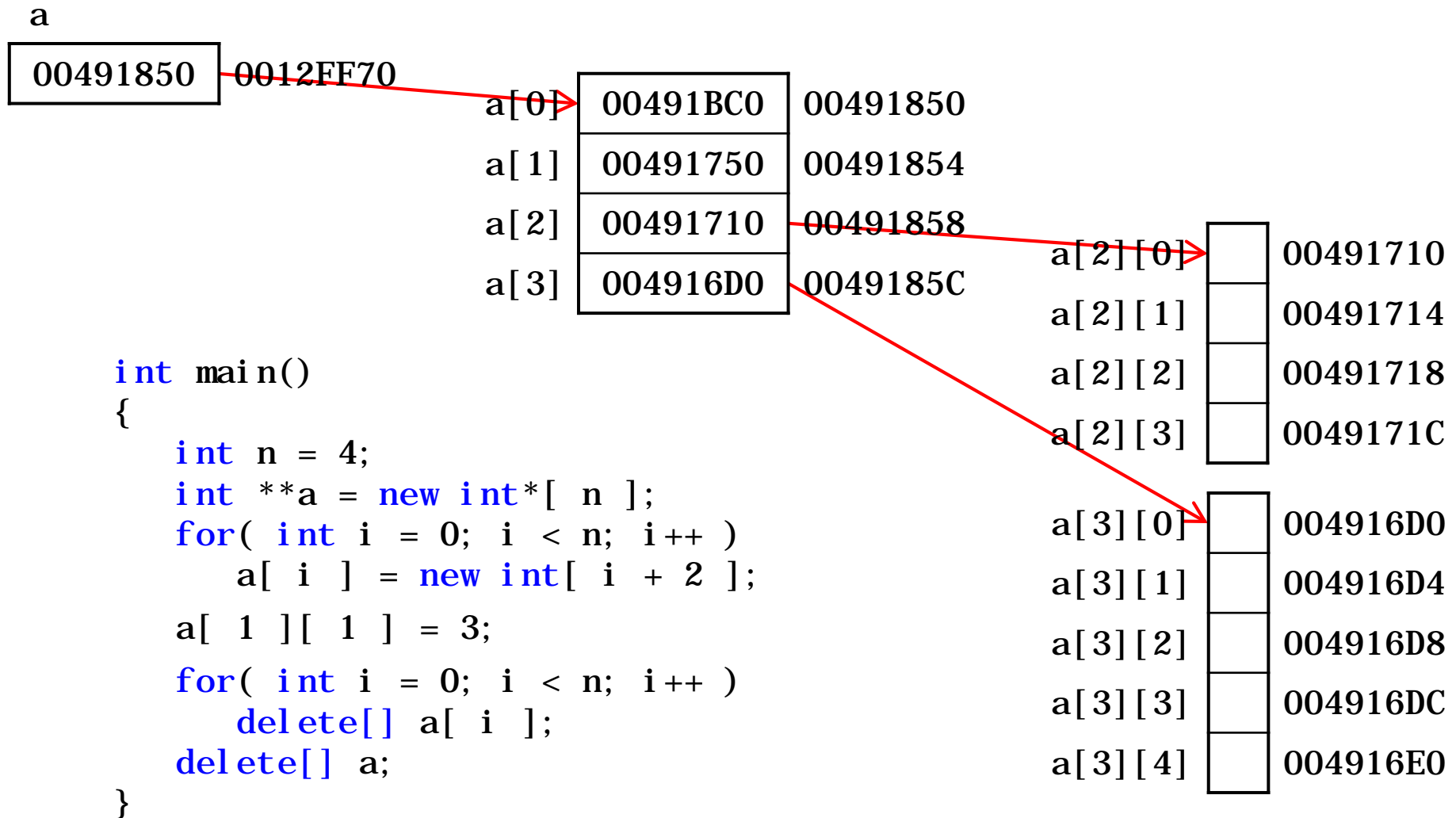


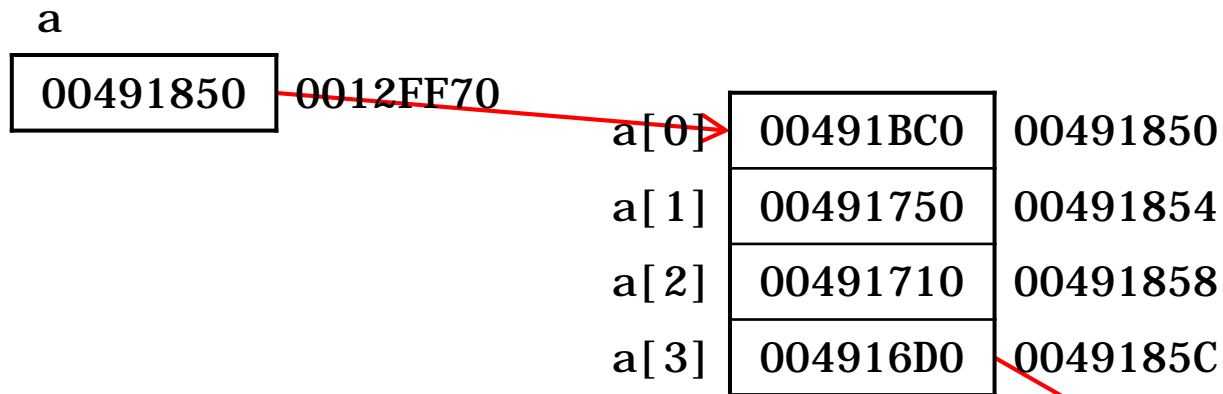
```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```



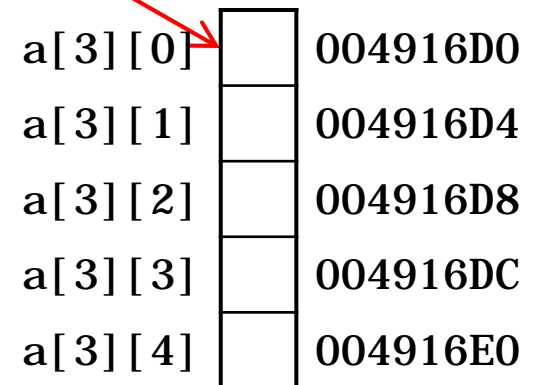
```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```

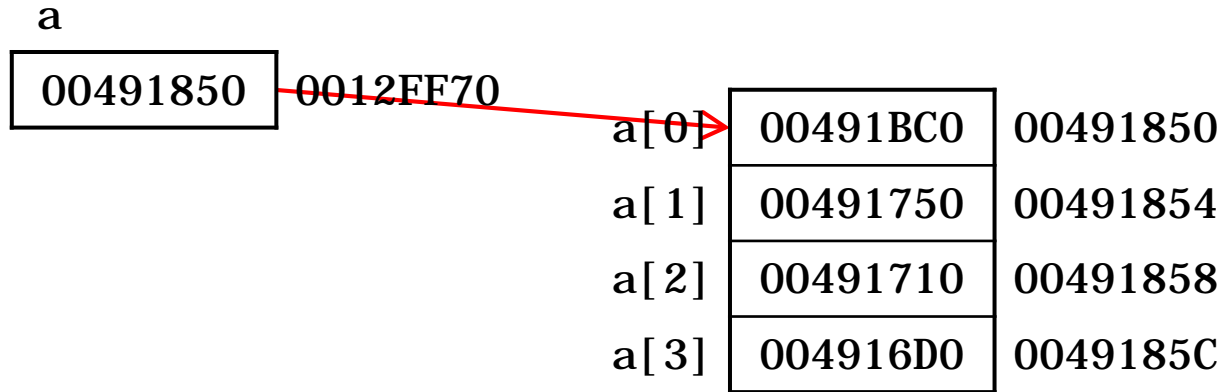






```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```





```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```

a

00491850
----------

 0012FF70

```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    a[ 1 ][ 1 ] = 3;
    for( int i = 0; i < n; i++ )
        delete[] a[ i ];
    delete[] a;
}
```

Passing dynamically allocated  
array of arrays to functions

```

int main()
{
    int n;
    cout << "Enter a positive integer: ";
    cin >> n;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];

    fun( a, n );

    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

```

```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

```

a



0012FF70

```

int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    fun( a, n );
    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}

```

```
void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}
```

a

00491850

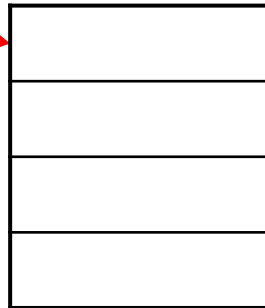
0012FF70

a[0]

a[1]

a[2]

a[3]



00491850

00491854

00491858

0049185C

```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    fun( a, n );
    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}
```



```
void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}
```

a

00491850 0012FF70

a[0]	00491BC0	00491850
a[1]		00491854
a[2]		00491858
a[3]		0049185C

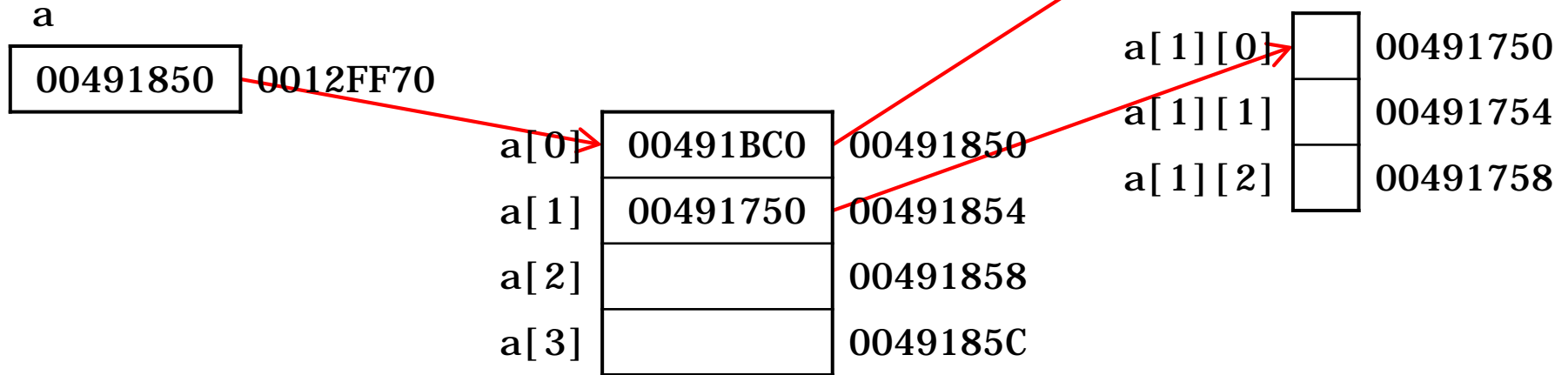
a[0][0] 00491BC0  
a[0][1] 00491BC4

```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    fun( a, n );
    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}
```

```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

```



```

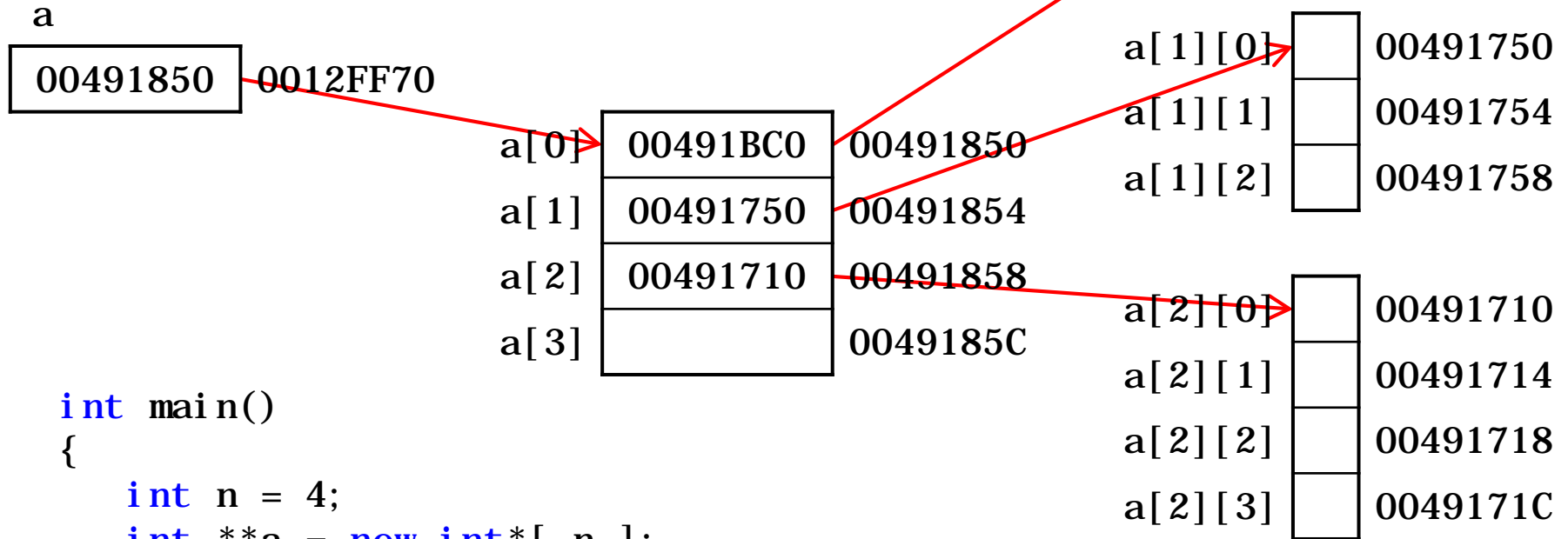
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    fun( a, n );
    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}

```

```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

```



```

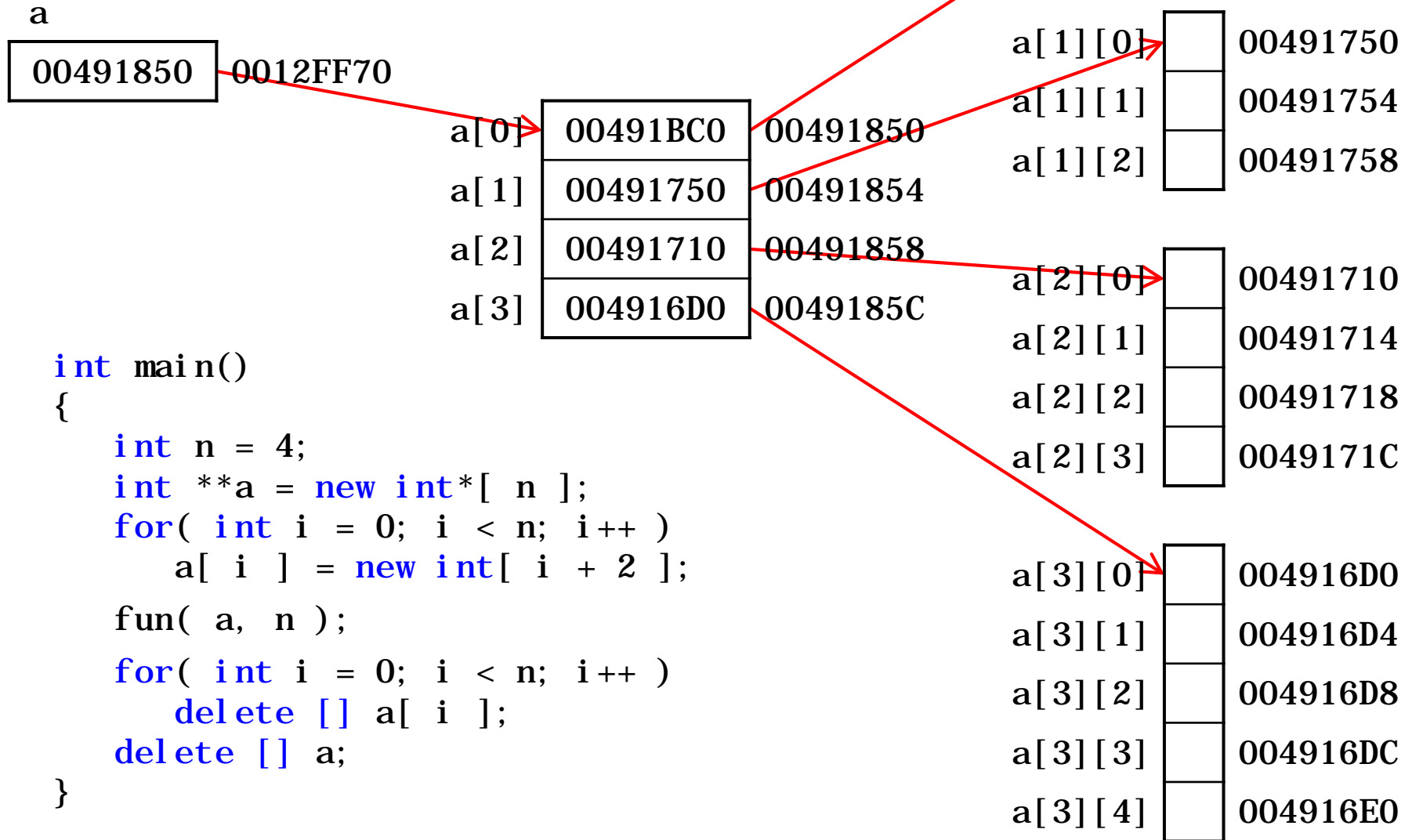
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    fun( a, n );
    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}

```

```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

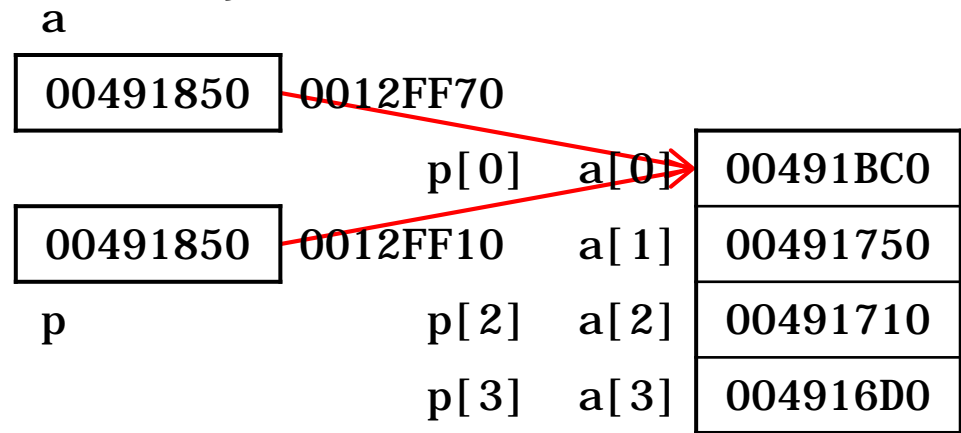
```



```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

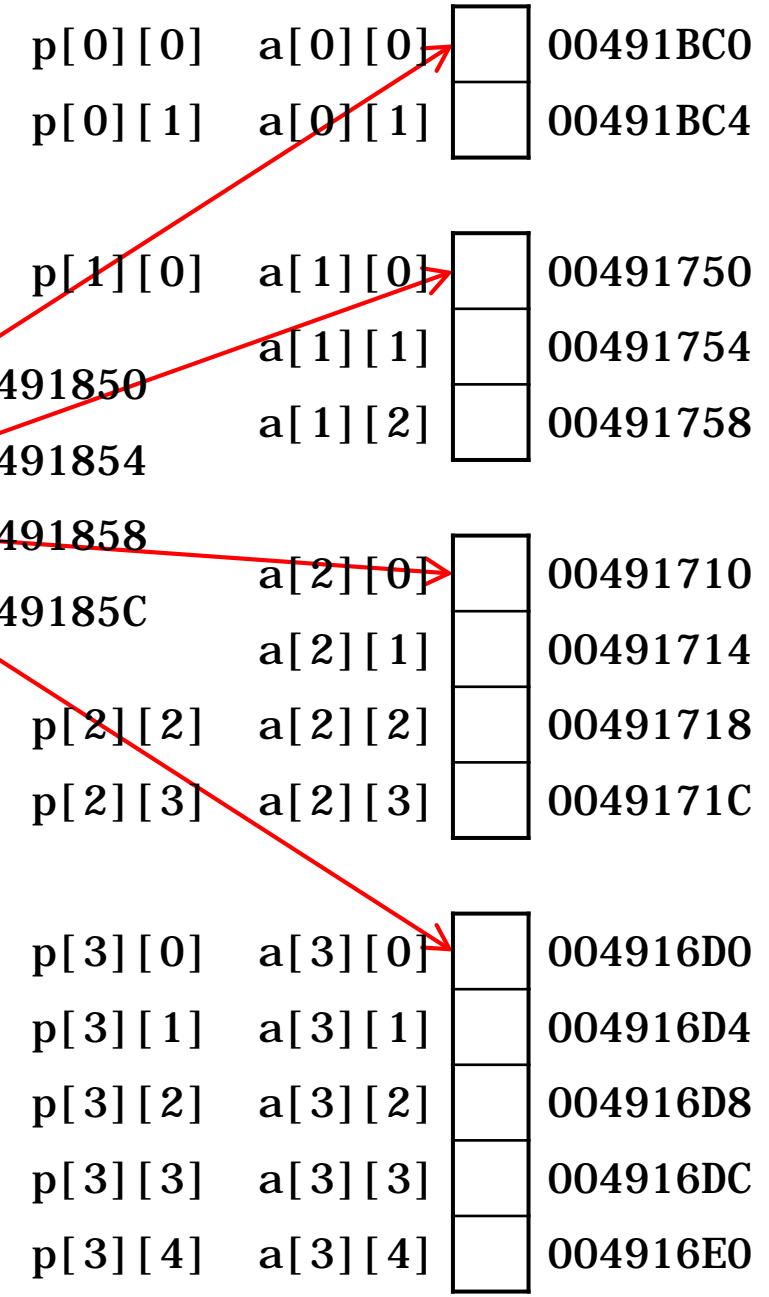
```



```

int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    fun( a, n );
    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}

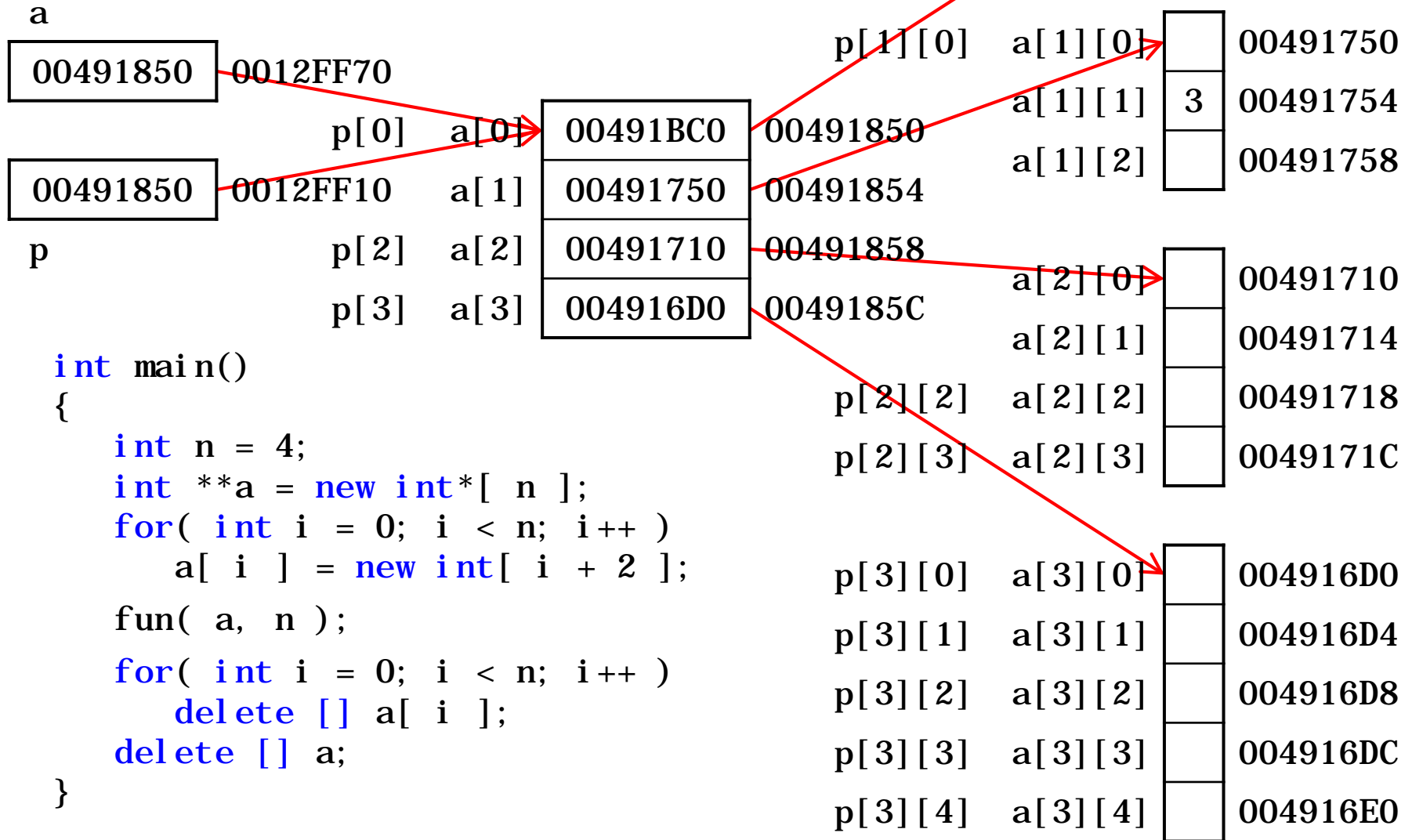
```



```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

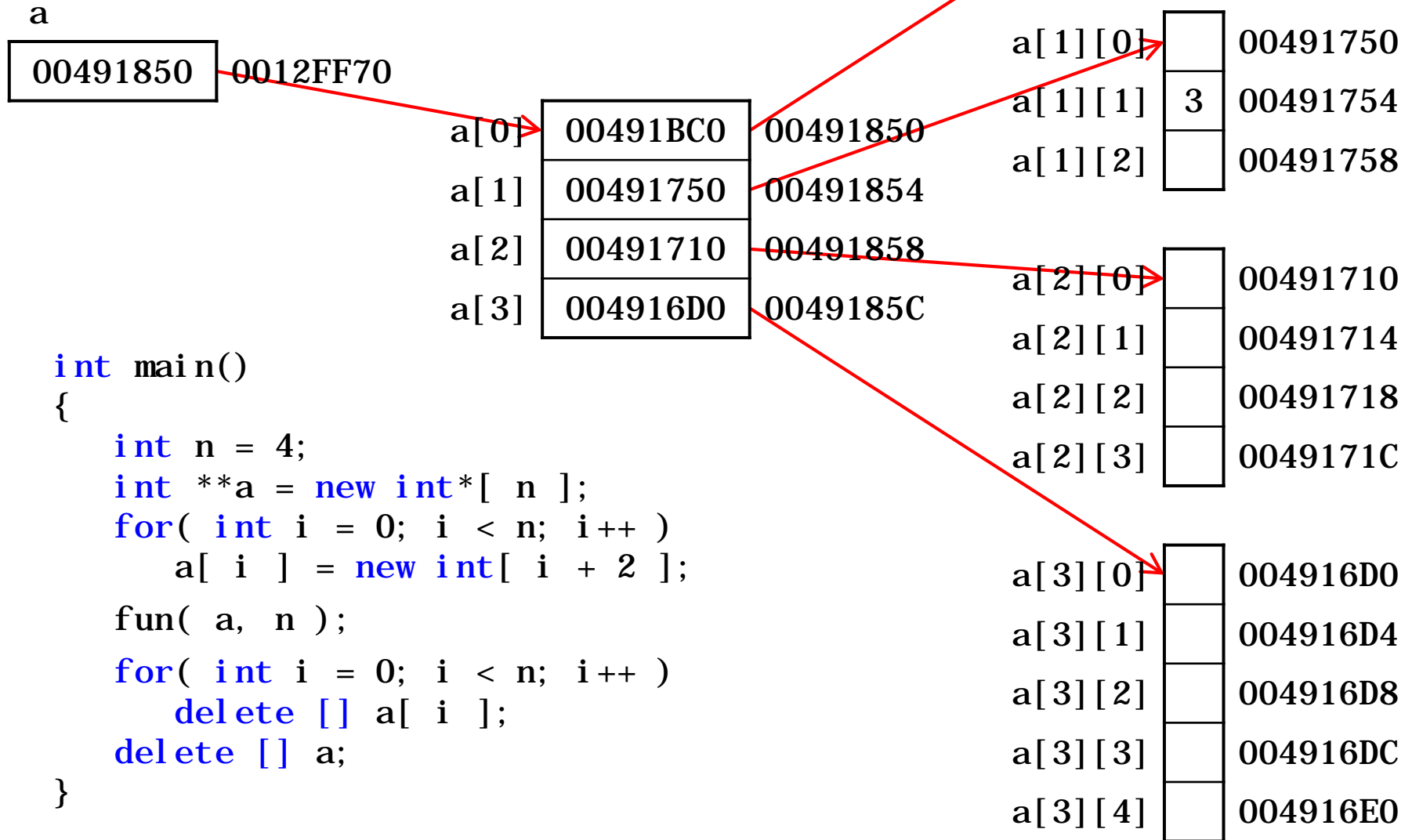
```



```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

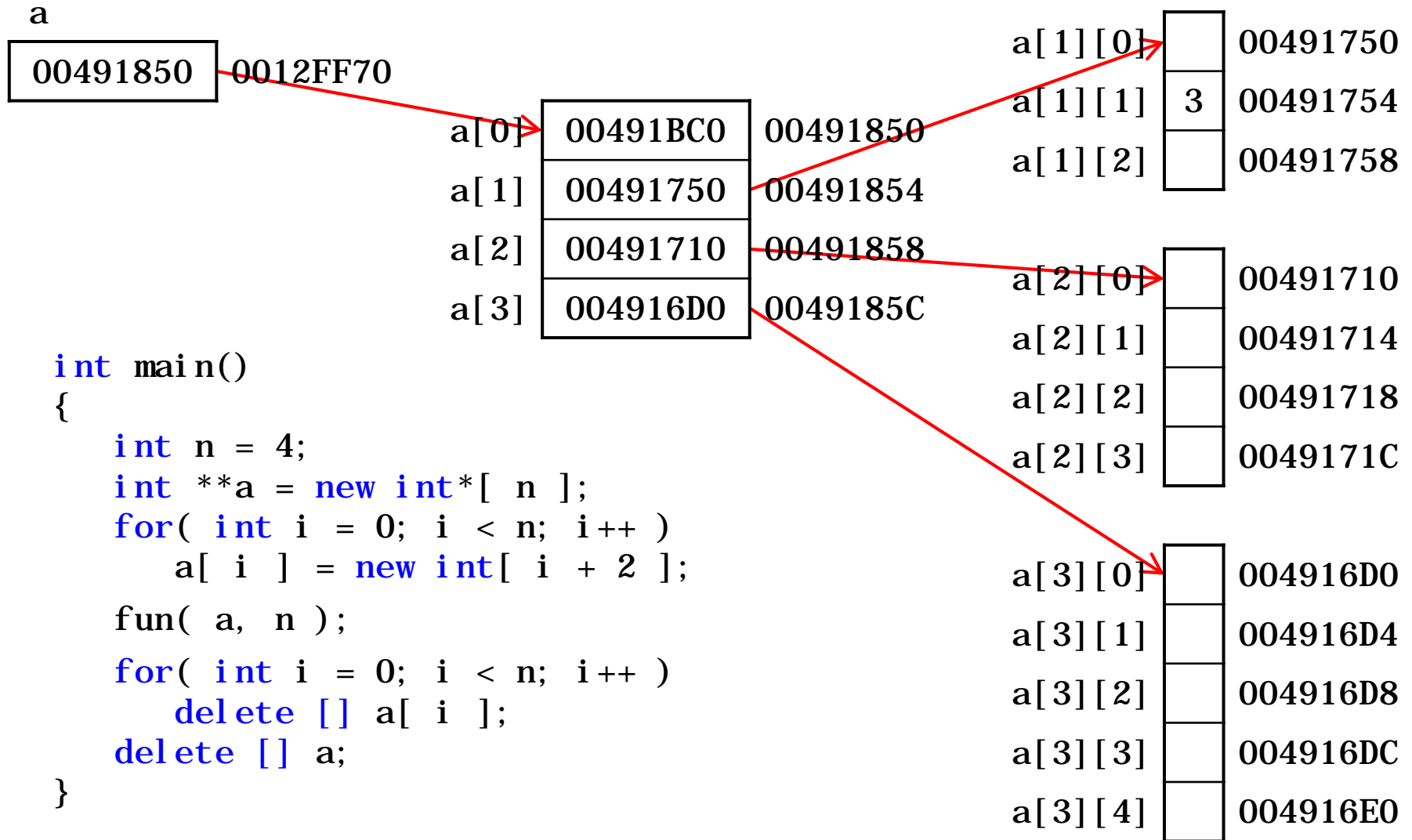
```



```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

```



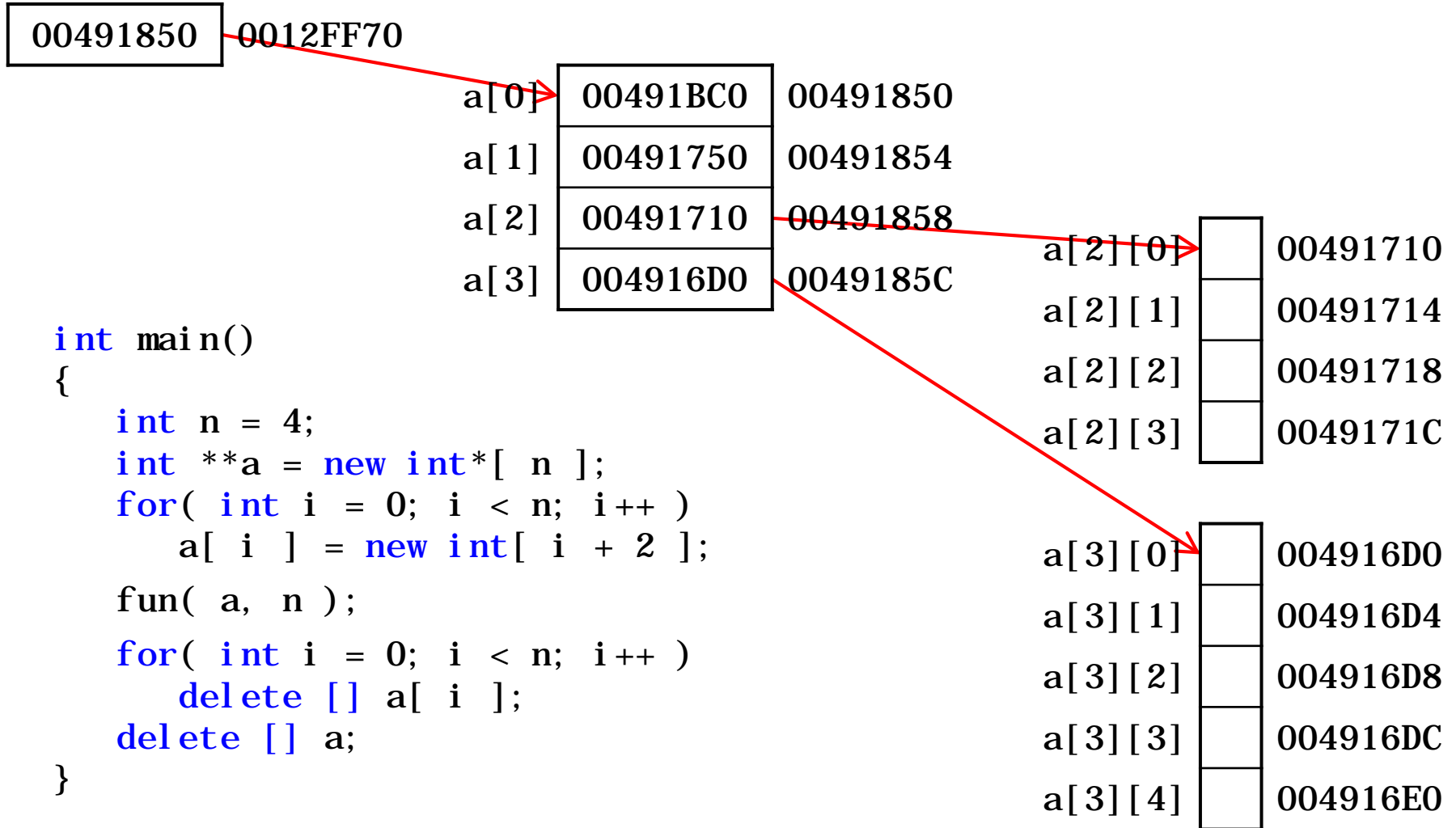


```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

```

a



```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

```

a

00491850

0012FF70

a[0]	00491BC0	00491850
a[1]	00491750	00491854
a[2]	00491710	00491858
a[3]	004916D0	0049185C

```

int main()
{

```

```

    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];

```

```

    fun( a, n );

```

```

    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;

```

```

}

```

a[3][0]		004916D0
a[3][1]		004916D4
a[3][2]		004916D8
a[3][3]		004916DC
a[3][4]		004916E0

```
void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}
```

a

00491850

0012FF70

a[0]	00491BC0	00491850
a[1]	00491750	00491854
a[2]	00491710	00491858
a[3]	004916D0	0049185C

```
int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    fun( a, n );
    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}
```

```

void fun( int **p, int n )
{
    p[ 1 ][ 1 ] = 3;
}

```

a

00491850	0012FF70
----------	----------

```

int main()
{
    int n = 4;
    int **a = new int*[ n ];
    for( int i = 0; i < n; i++ )
        a[ i ] = new int[ i + 2 ];
    fun( a, n );
    for( int i = 0; i < n; i++ )
        delete [] a[ i ];
    delete [] a;
}

```